



## INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 : <b>G06F 15/38</b>	A1	(11) International Publication Number: <b>WO 94/19755</b> (43) International Publication Date: 1 September 1994 (01.09.94)
(21) International Application Number: <b>PCT/US94/02136</b>		(81) Designated States: JP, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).
(22) International Filing Date: 28 February 1994 (28.02.94)		
(30) Priority Data: 08/023,976 26 February 1993 (26.02.93) US		<b>Published</b> <i>With international search report.</i> <i>Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>
(71) Applicant: MICROSOFT CORPORATION [US/US]; One Microsoft Way, Redmond, WA 98052-6399 (US).		
(72) Inventor: NEUMANN, Angelika, Gertrud, Claudia; 32 - 240th Street S.E., Bothell, WA 98021 (US).		
(74) Agents: CREAN, Timothy, J. et al.; Seed and Berry, 6300 Columbia Center, 701 Fifth Avenue, Seattle, WA 98104-7092 (US).		

(54) Title: METHOD AND SYSTEM FOR TRANSLATING DOCUMENTS USING TRANSLATION HANDLES

234

1404

DEMO.DOC

**Finding Text in a Document**

Wählen Sie aus dem Menü Bearbeiten den Befehl Suchen. In the Find What box<sup>J</sup>, type the text you want to find. Then choose the Find Next button<sup>K</sup> to begin the search.

**Replacing Text in a Document**

Wählen Sie aus dem Menü Bearbeiten den Befehl Ersetzen. Type the text you want to find in the Find What box<sup>J</sup>, and the replacement text in the Replace With box<sup>K</sup>. Then choose the Find Next button<sup>L</sup> to begin the search.

202

## (57) Abstract

A translation assistance program of a word processing program used to translate a source language document written in a source language to a target language document written in a target language. The translation assistance program displays the source language document on a translation screen portion of a display device. The translation assistance program then compares each source term in a product glossary with the source terms in the source language document. The product glossary associates each source term it stores with a target term which translates the source term into the target language. For each source term from the source language document which is stored in the product glossary, the translation assistance program: 1) underlines the source term in the source language document, 2) inserts a translation handle on the translation screen portion immediately after the source term, and 3) associates the translation handle with the target term which translates the source term into the target language. When the user decides to insert the target term at a current insertion point on the translation screen, the user invokes an insert target term command which contains the translation handle associated with the appropriate source term. In response, the translation assistance program retrieves the target term associated with the translation handle and inserts the target term at the current insertion point on the translation screen portion.

**FOR THE PURPOSES OF INFORMATION ONLY**

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	GB	United Kingdom	MR	Mauritania
AU	Australia	GE	Georgia	MW	Malawi
BB	Barbados	GN	Guinea	NE	Niger
BE	Belgium	GR	Greece	NL	Netherlands
BF	Burkina Faso	HU	Hungary	NO	Norway
BG	Bulgaria	IE	Ireland	NZ	New Zealand
BJ	Benin	IT	Italy	PL	Poland
BR	Brazil	JP	Japan	PT	Portugal
BY	Belarus	KE	Kenya	RO	Romania
CA	Canada	KG	Kyrgyzstan	RU	Russian Federation
CF	Central African Republic	KP	Democratic People's Republic of Korea	SD	Sudan
CG	Congo	KR	Republic of Korea	SE	Sweden
CH	Switzerland	KZ	Kazakhstan	SI	Slovenia
CI	Côte d'Ivoire	LI	Liechtenstein	SK	Slovakia
CM	Cameroon	LK	Sri Lanka	SN	Senegal
CN	China	LU	Luxembourg	TD	Chad
CS	Czechoslovakia	LV	Latvia	TG	Togo
CZ	Czech Republic	MC	Monaco	TJ	Tajikistan
DE	Germany	MD	Republic of Moldova	TT	Trinidad and Tobago
DK	Denmark	MG	Madagascar	UA	Ukraine
ES	Spain	ML	Mall	US	United States of America
FI	Finland	MN	Mongolia	UZ	Uzbekistan
FR	France			VN	Viet Nam
GA	Gabon				

DescriptionMETHOD AND SYSTEM FOR TRANSLATING DOCUMENTS  
USING TRANSLATION HANDLES

5

Technical Field

This invention relates generally to a computer method and system for automated translation of documents, and more specifically, to a computer method and system for 10 automated translation of documents using translation handles.

Background of the Invention

Due to the globalization of trade and the advent 15 of common markets, the process of converting information from one language to another with a computer, called machine-assisted translation (MAT), is becoming increasingly important.

In MAT systems a human translator prepares a 20 target language document, in a target language, from a source language document, in a source language, using a word processing program, automatic terminology managers, on-line multilingual term banks, and other computer-based tools which boost translator productivity.

An example using Figure 1 will help illustrate 25 how existing MAT systems operate. Figure 1 is a block diagram of a system 100 which assists a human translator in translating a source language document 112 to a target language document 116. The system 100 consists of a computer 102, a keyboard 104, a mouse 106, and a display screen 108.

A user of the system 100 invokes a word 35 processing program 110 using either the keyboard 104 or the mouse 106, and creates the source language document 112 containing the text "This is an example". The user then invokes a translation assistant 114 to assist the translator in translating the source language document 112

from the source language (e.g., English) to the target language document 116 containing the corresponding sentence in the target language (e.g., German).

To assist the user in translating the source language document 112 to the target language document 116 the translation assistant 114 displays a translation screen 118 and a dictionary screen 120 on the display screen 108. The translation screen 118 displays the text of the source language document 112. The dictionary screen 120 displays source terms 124 and corresponding translation options 126 which translate a given source term 124 into the target language. The translation assistant 114 retrieves each translation option 126 from a translation dictionary 122 which stores source terms 124 and their corresponding translation options 126. Each translation option 126 on the dictionary screen 120 is preceded by an identifier such as "a)", "b)", "c)" etc.

With the translation screen 118 and the dictionary screen 120 displayed on the display screen 108, the user reads the source language document 112 displayed on the translation screen 118. Next, the user looks away from the translation screen 118 and at the dictionary screen 120. By looking at the dictionary screen 120, the user finds the source terms 124 from the source language document 112 which are displayed on the dictionary screen 120 due to a match with the <source terms 124, translation option 126> pairs stored in the translation dictionary 122 of the translation assistant 114. The user then chooses which displayed source term 124 to translate. Then the user determines which corresponding translation option 126 to select to translate the chosen source term 124. Finally, the user invokes a copy translation command which copies the selected translation option 126 from the dictionary screen 120 to the translation screen 118. The user then looks away from the dictionary screen 120 to the translation screen 118 to see the selected translation option 126 displayed on the translation screen 118.

It is desirable to avoid looking from the translation screen 118 to the dictionary screen 120 in order to insert translation options 126.

It is also desirable to avoid looking from the  
5 translation screen 118 to the dictionary screen 120 to determine which source terms 124 from the source language document 112 have been matched with <source term 124, translation option 126> pairs from the translation dictionary 122 of the translation assistant 114.

10

#### Summary of the Invention

It is an object of the present invention to provide a function and system to ensure that consistent terminology is used throughout the target language  
15 document.

It is another object of the present invention to provide a function and system which offers typing aids to increase the translator's efficiency when translating a source language document into a target language document.

20

It is another object of the present invention to provide a function and system which allows translations of source terms to be inserted when and where they are needed in a sentence being translated.

25

It is another object of the present invention to facilitate the process of on-line translation by offering fast access to translated terminology without the need to look at the dictionary screen.

30

These and other objects, which will be apparent as the invention is more fully described, are provided by a function and system for translating a source language document to a target language document using translation handles. In the preferred embodiment of the present invention, a user invokes a translation assistance program under a word processing program to translate a source  
35 language document written in a source language to a target language document written in a target language. The translation assistance program displays the source

language document on a translation screen of a display device. The translation assistance program then compares each source term in a product glossary with the source terms in the source language document. The product 5 glossary associates each source term it stores with a target term which translates the source term into the target language. For each source term from the source language document which is found in the product glossary, the translation assistance program: 1) underlines the 10 source term in the source language document, 2) inserts a translation handle on the translation screen immediately after the underlined source term, and 3) associates the translation handle with the target term which translates the source term into the target language. When the user 15 decides to insert the target term at a current insertion point on the translation screen, the user invokes an insert target term command which contains the translation handle associated with the appropriate source term. In response, the translation assistance program retrieves the 20 target term associated with the translation handle and inserts the target term at the current insertion point on the translation screen.

Brief Description of the Drawings

25 Figure 1 is a block diagram showing a prior art system for translating a source language document to a target language document.

Figure 2 is a block diagram of a system embodying the present invention for translating a source 30 language document to a target language document.

Figure 3 is a block diagram of typical Microsoft Lingua categories.

Figure 4 is a flow diagram of the function CreateProductGlossary for the system of Figure 2.

35 Figure 5 is a flow diagram of the function SortProductGlossary for the system of Figure 2.

Figure 6 is a flow diagram of the function PrepareDocumentForTranslation for the system of Figure 2.

Figure 7 is a flow diagram of the function TagEachParagraph for the system of Figure 2.

5 Figure 8A is a flow diagram of the function SearchForMatch for the system of Figure 2.

Figure 8B is a flow diagram of the function MatchStaticString for the system of Figure 2.

10 Figure 8C is a flow diagram of the function MatchTemplateString for the system of Figure 2.

Figure 9 is a flow diagram of the function GenerateTranslationHandle for the system of Figure 2.

Figure 10 is a flow diagram of the function InsertTranslation for the system of Figure 2.

15 Figure 11 is a flow diagram of the function ToggleFirstCharacter for the system of Figure 2.

Figure 12 is a flow diagram of the function UndoTranslation for the system of Figure 2.

20 Figure 13 is a flow diagram of the function CreateTranslationParagraph for the system of Figure 2.

Figure 14A is a block diagram of the source language document after it has been prepared for translation by the system of Figure 2.

25 Figure 14B is a block diagram of the source language document just before a target term will be inserted in the source language document using the insert target term command.

Figure 15A is a block diagram of the source language document after the user manually types in the 30 translation for the rest of the heading.

Figure 15B is a block diagram of the source language document after the user deletes a source language heading.

35 Figure 16A is a block diagram of the source language document after the system of Figure 2 has automatically inserted a translation for a source language sentence.

Figure 16B is a block diagram of the source language document after the system of Figure 2 has inserted a target term in the source language document in response to the insert target term command.

5 Figure 17A is a block diagram of the source language document after the user has mistakenly inserted the wrong target term using the insert target term command.

10 Figure 17B is a block diagram of the source language document after the user has invoked an undo translation handle request.

Figure 18A is a block diagram of the source language document.

15 Figure 18B is a block diagram of the source language document after the user has entered the insert target term command for a template string.

Figure 19A is a block diagram of the source language document after the user has issued a toggle capitalization request.

20 Figure 20A is a block diagram of the source language document after a source language paragraph has been deleted by the paragraph command.

25 Figure 20B is a block diagram of the target language document created after the source language document has been fully translated using the system of Figure 2.

Figure 21 is a block diagram of a product glossary for the system of Figure 2 used to implement the translations depicted in Figures 14A-20B.

30 Appendix A is a user's guide for the system of Figure 2.

Detailed Description of the InventionOverview:

The present invention provides a function and system which overcomes the problems of existing machine-assisted translation (MAT) systems. The preferred system 5 200 (Figure 2) of the present invention provides a translation assister 208 which translates a source language document 202, written in a source language, to a target language document 206 written in a target language. 10 First, the translation assister 208 displays the source language document 202 on a display device 234. The translation assister 208 then compares each source term 214 in a product glossary 212 with each source term 203 in the source language document 202. The product glossary 15 212 associates each source term 214 with a target term 218 which translates the source term 214 into the target language. For each source term 214 from the source language document 202 which matches at least one source term 214 in the product glossary 212, the translation 20 assister 208: 1) underlines the source term 203 in the source language document 202, 2) inserts a translation handle (described in more detail below) such as a superscript character, immediately after the source term 203, and 3) associates the translation handle with the 25 target term 218 in the product glossary 212 which translates the source term 203 into the target language.

When the user decides to translate the source term 203 by inserting the target term 218 on the display device 234, the user of the system 200 invokes an insert 30 target term command which contains the translation handle associated with the source term 203 to be translated. In response, the translation assister 208 retrieves the target term 218 associated with the translation handle contained in the insert target term command, and inserts 35 the target term 218 on the display device 234.

The System:

An example using Figure 2 will illustrate a typical translation process using a MAT system 200 operating according to the present invention. First, a user of the system 200 invokes word processing program 204 using a mouse 205. Then the user enters text on a keyboard 207 to create the source language document 202. Both the word processing program 204 and the source language document 202 are stored in the computer memory 209. The source language document 202 comprises a series of source terms 203 from a source language (e.g., English). Once the source language document 202 has been created, the user often desires to translate the source language document 202 into the target language document 206 (e.g., German).

To perform this translation, the present invention provides, in addition to the word processing program 204, a translation assister 208 and a spreadsheet program 210. The user invokes the spreadsheet program 210 and creates a product glossary 212 which associates a source term 214 with a target term 218 which is a translation of the source term 214 into the target language. The preferred product glossary 212 contains four columns of data: the source term 214, a category 216, the target term 218, and a comment 220. In the preferred embodiment, the order of the columns of the product glossary 212 must be the same order shown in the product glossary 212 of Figure 2, and the four columns must be named "Source Term", "Category", "Target Term", and "Comment".

The category 216 classifies each <source term 214, target term 218> pair. The product glossary 212 preferably contains categories 216 which conform to standard Microsoft Lingua abbreviations. Figure 3 sets forth some typical Lingua abbreviations such as "MNU" for menu names, "COM" for command names, and "MAC" for macro names.

In the preferred embodiment the source term 214 and the target term 218 comprise a word, a phrase, a sentence, or a template string. The template string is a commonly occurring sentence or phrase with embedded 5 template variables. Template variables are place holders for terms which change or vary. In the preferred embodiment template variables are identified by a leading % sign followed by the appropriate Lingua abbreviation (such as one of those shown in Figure 3). Thus, the 10 template variable for a menu name is %MNU.

A brief discussion of the invention as it applies to the template strings will help illustrate the usefulness of template strings. Assume a first phrase "Choose Open from the File menu" as well as a second 15 phrase "Choose Convert Table to Text from the Table menu" are commonly occurring phrases in the source language documents to be processed by the system 200. Through the use of template strings with embedded template variables, the system 200 can provide machine assistance in 20 translating these phrases to the target language.

For example, the user creates the template string for these commonly occurring phrases by first determining that the words "Choose \_\_\_\_\_ from the \_\_\_\_\_" are common to both of the phrases. Next, the 25 user determines that the word "Open" from the first phrase and the words "Convert Table to Text" from the second phrase are classified as commands (%COM) in the Lingua classification system. Then, the user determines that the word "File" in the first phrase and the word "Table" in 30 the second phrase are classified as menu names (%MNU) in the Lingua classification system. After making these determinations, the user invokes the spreadsheet program 210 and stores the following data in the product glossary 212:

TABLE 1

<u>SOURCE TERM</u>	<u>CATEGORY</u>	<u>TARGET TERM</u>
Choose %COM from the %MNU	TEM	Wählen Sie den Befehl %COM aus dem Menü %MNU
5 Convert Table to Text	COM	Tabelle in Text Umwandeln
10 Table	MNU	Tabelle
Open	COM	Öffnen
File	MNU	Datei
15 When the system 200 processes the source language document 202 for translation, the system 200 will recognize the phrase "Choose Open from the File menu" as well as the phrase "Choose Convert Table to Text from the Table menu" because both phrases match the template string set forth in Table 1. Upon recognition of either phrase, the system 200 provides easy access to a translation for the phrase, using the target term associated with the template string stored in Table 1. The process used by the system 200 to provide such easy access to the 20 translation for each phrase is described in more detail below.		
25		

Returning to the discussion of how to create the preferred product glossary 212 using the spreadsheet program 210, the user invokes the spreadsheet program 210 and stores appropriate data in the product glossary 212. The user of the system 200 then invokes a product glossary sorter 222 of the translation assister 208 to sort the data in the product glossary 212. In the preferred embodiment, the product glossary sorter 222 performs the 30 steps of the method SortProductGlossary (set forth in more detail below in Figure 5). The product glossary sorter 222 retrieves all source terms 214 which are template strings and sorts the template strings according to length, with the longest source term 214 being stored in 35 the product glossary 212 before the shortest source term. 40

For template strings which are of equal length, the product glossary sorter 222 invokes a product glossary alphabetizer 224 which alphabetizes the template strings which are of equal length. Any well-known alphabetizing program can be used to implement the product glossary alphabetizer 224.

Next, the product glossary sorter 222 sorts by length all source terms 214 which are not template strings. Then the product glossary alphabetizer 224 case-sensitively alphabetizes the non-template strings which are of equal length. In this way upper case versions of a letter are sorted above lowercase versions of the letter.

After the product glossary alphabetizer sorter 224 properly sorts the <source term 214, target term 218, category 216> tuples of the product glossary 212, the user invokes a source language document preparer 226 of the translation assister 208 which prepares the source language document 202 for translation by the user of the system 200. In the preferred embodiment, the source language document preparer 226 performs the steps of the function PrepareDocumentForTranslation set forth in more detail below and shown in Figure 6.

To prepare the source language document 202 in a manner to assist the user in translating the source language document 202 into the target language document 206, the source language document preparer 226 invokes a tag paragraph program 228, a comparer 230, and a translation handle generator 232.

The tag paragraph program 228 creates a paragraph tag, which uniquely identifies a paragraph within the source language document 202, for each paragraph in the source language document. In the preferred embodiment, the tag paragraph program 228 performs the steps of the function TagEachParagraph set forth in more detail below and shown in Figure 7.

The source language document preparer 226 then invokes the comparer 230. In the preferred embodiment,

the comparer 230 performs the steps of the function SearchForMatch set forth in more detail below and shown in Figures 8A-8C. The comparer 230 compares each source term 214 in the product glossary 212 with the source terms 203 5 in the source language document 202. For each "match" between one of the given source terms 214 in the product glossary 212 and one of the source terms 203 in the source language document 202, the source language document preparer 226 invokes the translation handle generator 232.

10 The translation handle generator 232 associates the matched source term 203 with the target term 218 which translates the source term 203 into the target language. In the preferred embodiment, the translation handle generator 232 performs the steps of the function 15 GenerateTranslationHandle set forth in more detail below and shown in Figure 9.

The translation handle generator 232 first underlines the matched source term 203 so as to make the matched source term 203 easier to locate when displayed on 20 a display device 234 than unaltered source terms. The translation handle generator 232 then inserts a translation handle in the form of a superscript character (not shown) immediately after the underlined source term 203 (e.g., the letter J).

25 For example, if the source language document 202 contains the phrase "Finding Text in a Document" and the product glossary 212 contains the source term "Document", then the comparer 230 will find a "match" between the source term "Document" in the source language document 202 30 and the source term "Document" in the product glossary 212. The translation handle generator 232 will then alter the appearance of the phrase "Finding Text in a Document" so that it appears as "Finding Text in a Document<sup>J</sup>". Returning to the discussion of the translation handle 35 generator 232, the translation handle generator 232 stores in an index file 248 a paragraph tag 250, a translation handle 252, a source term 254, and a target term 256.

The paragraph tag 250 identifies the paragraph which contains the underlined source term 203. The translation handle 252 corresponds to the superscript character inserted after the underlined source term 203. The source term 254 corresponds to the underlined source term 203. The target term 256 corresponds to the target term 218 from the product glossary 212 which translates the underlined source term 203 into the target language.

In this way, the translation generator 232 associates the translation handle 252 with the target term 256 which translates the source term 254 into the target language. This association allows a user of the system 200 to retrieve and insert the target term 256 merely by invoking an insert target term command containing the appropriate translation handle 252.

Thus the translation handle generator 232 does not typically insert a target term 256 directly in place of a matched source term 203 because such an insertion only saves time if it happens to occur at the beginning or 20 the end of a source language sentence in the source language document 202, and if the syntax of an equivalent target language sentence is the same as that of the source language sentence 258. By associating the translation handle 252 with the target term 256 in the index file 248 25 the present invention provides access to translations which the user inserts when and where they are needed in the source language sentence through the use of insert target term commands.

Once the source language document 202 has been 30 prepared by the source language document preparer 226, the user translates the source language document 202 into the target language document 206 using a translation inserter 236, a translation paragraph creator 238, a character changer 240, and a translation remover 242, all of which 35 will be discussed in more detail below in connection with the translation assister 208.

First, the user invokes the word processing program 204 to display source terms 235 in the prepared source language document 202 on the display device 234. The source terms 235 on the display device 234 correspond 5 to the source terms 203 of the source language document 202 stored in the computer memory 209. When the user determines that the target term 256 associated with the translation handle 252 should be inserted at a current insertion point displayed on the display device 234, then 10 the user invokes the insert target term command using the keyboard 207 or the mouse 205. The insert target term command automatically invokes the translation inserter 236. In the preferred embodiment, the insert target term command consists of the simultaneous depression using the 15 keyboard 207 of a CTRL key (not shown), a SHIFT key (not shown) and a keyboard 207 letter key (not shown) associated with the translation handle 252. The process by which the system 200 carries out the insert target term command is set forth in more detail below and is shown in 20 Figure 10.

In response to the insert target term command, the translation inserter 236 retrieves the target term 256 from the index file 248 which is associated with the superscript character contained in the insert target term 25 command and the paragraph which currently contains the insertion point, and inserts the target term 256 at the current insertion point on the display device 234. In the preferred embodiment, the translation inserter 236 performs the steps of the function InsertTranslation set 30 forth in more detail below and shown in Figure 10.

The present invention also allows the user to insert target terms 256 in a separate translation paragraph (not shown) or in the source language sentence 237, on the display screen 234, to be translated. To 35 insert target terms 256 in the source language sentence 237, the user merely invokes the insert mode of the word processing program 204. Typically this involves

depressing an insert key on the keyboard 255. As target terms 256 are inserted in the source language sentence 237, the source terms 235 in the source language sentence 237 change position on the display device 234 to provide  
5 space in which to insert the target terms 256.

To insert target terms 256 in a translation paragraph, the user enters a create translation paragraph command, which invokes the translation paragraph creater 238 of the translation assister 208. In the preferred  
10 embodiment, the translation paragraph creater 238 performs the steps of the function TranslationParagraph set forth in more detail below and shown in Figure 13. The advantage to the user of inserting target terms 256 in the separate translation paragraph is that the source terms  
15 235 on the display device 234 remain stationary during translation because they are in a separate paragraph.

The present invention also provides the character changer 240 which toggles the first character of the last target term 256 inserted with the insert target term command on the display device 234. In other words,  
20 if the first letter of the last inserted target term 256 is a lower case letter, then the character changer 240 replaces the lowercase letter with an uppercase version of the letter. Likewise, if the first letter of the target  
25 term 256 is an uppercase character, then the character changer 240 replaces the uppercase letter with a lowercase version of the uppercase letter. In the preferred embodiment, the character changer 240 performs the steps  
30 of the function ToggleFirstCharacter set forth in more detail below and shown in Figure 11.

The translation remover 242 of the translation assister 208 deletes the last target term 256 inserted on the display device 234 along with any text entered on the display device 234 between the last target term 256 and  
35 the current position of the insertion point. In the preferred embodiment, the translation remover 242 performs

the steps of the function UndoTranslation set forth in more detail below and shown in Figure 12.

Flow Diagrams

Figure 4 is a flow diagram of the function CreateProductGlossary which sets forth the manual steps performed by the user to create the product glossary 212 of Figure 2. In the preferred embodiment, the function CreateProductGlossary is performed using the spreadsheet program 210 (Figure 2). In step 401 of Figure 4, the function CreateProductGlossary creates a spreadsheet file with column headings, from left to right, of <source term, category, target term, comment>. In step 402, the function enters in the spreadsheet columns, the character strings typically found in the source language documents to be translated. In step 403, the function CreateProductGlossary ends processing.

Figure 5 is a flow diagram of the function SortProductGlossary which sorts the product glossary 212 according to the length of the source terms 214. In the preferred embodiment, the function SortProductGlossary is performed by the product glossary sorter 222 and the product glossary alphabetizer 224 (Figure 2). For all source terms 214 of equal length, the function case-sensitively alphabetizes the source terms 214.

In step 501 of Figure 5 the function creates four new column headings in the product glossary 212. The new column headings are <source term, category, target term, comment>. In step 502, the function retrieves all source terms 214 from the previously existing columns of data in the product glossary 212 which are template strings. In step 503, the function sorts all retrieved template strings by length, with the longest template strings sorted to the top of the file and above the shortest template strings. The function stores the sorted template strings in the new spreadsheet columns. In step 504, the function alphabetizes, in the new columns of the

product glossary 212, all template strings of equal length.

In step 505, the function sorts by length, all source terms 214 which are not template strings. In step 5 506, for all non-template source terms 214 of equal length, the function case-sensitively alphabetizes the non-template source terms 214, such that uppercase letters are sorted above lowercase letters in the product glossary 212. In step 507, the function deletes the "old" 10 spreadsheet columns where the original data was stored. In step 508, the function SortProductGlossary removes all duplicate entries from the product glossary 212. In step 509, the function SortProductGlossary returns processing control to the word processing program 204 (Figure 2).

15 Figure 6 is a flow diagram of the function PrepareDocumentForTranslation which underlines and assigns a translation handle to each source term 203 of the source language document 202 which has a corresponding source term 214 in the product glossary 212. In the preferred 20 embodiment, the function PrepareDocumentForTranslation is performed by the source language document preparer 226 (Figure 2).

In step 601 of Figure 6, the function retrieves 25 the product glossary 212. In step 602, the function PrepareDocumentForTranslation calls the function TagEachParagraph, which creates a paragraph tag for each paragraph in the source language document 202, that uniquely identifies that paragraph. The function TagEachParagraph is set forth in more detail below and 30 shown in Figure 7.

In step 603, the function determines whether unprocessed source terms 203 from the source language document 202 still remain. If unprocessed source terms 203 do still remain then the function performs steps 604 35 through 606. In step 604, the function PrepareDocumentForTranslation calls the program SearchForMatch which compares the source term 203 from the

source language document 202 with each source term 214 in the product glossary 212, while looking for a match between the source terms. The function SearchForMatch is set forth in more detail below and shown in Figures 8A, 5 8B, and 8C.

In step 605 the function PrepareDocumentForTranslation determines if a match was found by the function SearchForMatch, between the source term 203 and one of the source terms 214 of the product 10 glossary 212. If no match was found then the function PrepareDocumentForTranslation continues processing with step 603.

If a match was found between the source term 203 and one of the source terms 214 of the product glossary 15 212, then the function PrepareDocumentForTranslation calls the function GenerateTranslationHandle, in step 606. The function GenerateTranslationHandle underlines the matched source term 205, assigns a translation handle to the matched source term 203, and associates the translation 20 handle with the target term from the target language which translates the source term 203 into the target language. The function GenerateTranslationHandle is set forth in more detail below and shown in Figure 9. Upon completion 25 of step 606, the function PrepareDocumentForTranslation continues processing by returning to step 603.

Returning to the discussion of step 603, if all source terms 203 from the source language document 202 have been processed, then in step 607 the function PrepareDocumentForTranslation returns processing control 30 to the word processing program 204 (Figure 2).

Figure 7 is a flow diagram of the function TagEachParagraph which associates a paragraph tag, uniquely identifying each paragraph in the source language document 202, with a given paragraph. In the preferred 35 embodiment, the function TagEachParagraph is performed by the paragraph marker 228 (Figure 2).

In step 701 of Figure 7 the function TagEachParagraph determines whether every paragraph in the source language document 202 has been assigned a paragraph tag. If paragraphs remain in the source language document 202 which have not been assigned a paragraph tag, then the function TagEachParagraph continues processing with steps 702-704. In step 702 the function TagEachParagraph retrieves the first paragraph in the source language document 202 that has not been assigned a paragraph tag. In step 703 the function creates a paragraph tag by concatenating the name of the source language document 202 with a current paragraph number. The function assigns paragraph numbers sequentially from a first paragraph in the source language document 202 to a last paragraph in the source language document. In step 704 the function TagEachParagraph associates the paragraph tag with the retrieved paragraph. In a preferred embodiment the paragraph tag is implemented as a paragraph mark in Microsoft Word Version 2.0. Paragraph tags are described in more detail in the "Word for Windows 2.0 User's Guide", Chapter 40, and "Using Word Basic", both of which are incorporated herein by reference. Upon completion of step 704, the function continues processing with step 701.

Returning to the discussion of step 701, if the function TagEachParagraph determines that every paragraph in the source language document 202 has been assigned a paragraph tag, then the function TagEachParagraph returns processing control to the function PrepareDocumentForTranslation.

Figure 8A is a block diagram of the function SearchForMatch which determines whether a match exists between the source terms 214 from the product glossary 212 and any of the source terms 203 from the source language document 202. In the preferred embodiment the function SearchForMatch is performed by the comparer 230 (Figure 2).

In step 801 of Figure 8A the function determines if there are source terms 214 from the product glossary 212 that have not yet been processed. If at least one unprocessed source term 214 remains in the 5 product glossary 212 then in step 802 the function retrieves the first unprocessed source term 214 from the product glossary 212. In step 803 the function determines whether the retrieved source term 214 is a template string. If the retrieved source term 214 is a template 10 string then in step 804 the function SearchForMatch calls the function MatchTemplateString. Upon completion of step 804 the function SearchForMatch returns processing control to the function PrepareDocumentForTranslation (Figure 6). The function MatchTemplateString is set forth 15 in more detail below and is shown in Figure 8C.

Returning to the discussion of step 803, if the retrieved source term 214 is not a template string then the function SearchForMatch, in step 806, calls the function MatchStaticString set forth in more detail below 20 and shown in Figure 8B. Upon completion of step 806 the function SearchForMatch returns processing control to the function PrepareDocumentForTranslation (Figure 6).

Figure 8B is a flow diagram of the function MatchStaticString which compares non-template source 25 terms 214 retrieved from the product glossary 212 with the source terms 203 and the source language document 202. In the preferred embodiment the comparer 230 (Figure 2) performs the steps of the function MatchStaticString.

In step 807 of Figure 8B the function 30 MatchStaticString compares the retrieved source term 214 from the product glossary 212 with the source terms 203 in the source language document 202. In step 808 the function determines whether there is a match between the retrieved source term 214 from the product glossary 212 35 and at least one of the source terms 203 from the source language document 202. If there was a match then processing continues with steps 809-814. In step 809 the

function determines if there is a one-to-one correspondence between the retrieved source term 214 from the product glossary 212 and the matched source terms 203 in the source language document 202. If a one-to-one 5 correspondence exists then in step 810 the function assigns the value true to the boolean variable match. In step 811 the function MatchStaticString returns processing control to the function SearchForMatch (Figure 8A).

Returning to the discussion of step 809, if the 10 function determines that there is not a one-to-one correspondence then in step 812 the function determines if there are less than five letters appended to the end of the source terms 203 from the source language document 202 which do not match the source term 214 retrieved from the 15 product glossary 212. If the "extra" letters are less than five, then in step 813, the function assigns the value true to the boolean variable match. Upon completion of step 813, the function MatchStaticString, in step 811, returns processing control to the function SearchForMatch 20 (Figure 8A).

Returning to the discussion of step 812, if the function determines that there are five or more letters appended to the end of the source term(s) 203 from the source language document 202, then in step 814, the 25 function assigns the value false to the boolean variable match. Upon completion of step 814 the function MatchStaticString returns processing control, in step 811, to the function SearchForMatch (Figure 8A).

Returning to the discussion of step 808, if 30 there was not a match between the retrieved source term 214 from the product glossary 212 and any of the source terms 203 in the source language document 202 then the function, in step 815, assigns the value false to the boolean variable match. Upon completion of step 815 the 35 function MatchStaticString, in step 811, returns processing control to the function SearchForMatch (Figure 8A).

Figure 8C is a flow diagram of the function MatchTemplateString which compares the template string retrieved from the product glossary 212 with the source terms 203 from the source language document 202. In the 5 preferred embodiment the comparer 230 (Figure 2) performs the steps of the function MatchTemplateString.

In step 816 of Figure 8C the function assigns the value true to the boolean variable match. In step 817 the function assigns a position indicator to the beginning 10 of the source language document 202. In step 818 the function determines if there is a portion of the template string that remains unprocessed. If the entire template string has been processed then in step 819 the function MatchTemplateString returns processing control to the 15 function SearchForMatch (Figure 8A).

If an unprocessed portion of the template string remains, then in step 820 the function MatchTemplateString determines if the variable match is currently set to true. If the variable match is currently set to false then in 20 step 821 the function MatchTemplateString returns processing control to the function SearchForMatch (Figure 8A).

Returning to the discussion of step 820, if the variable match is set to true then in step 822 the 25 function MatchTemplateString determines if the unprocessed portion of the template string is a template variable. If the function determines that the unprocessed portion of the template string was a template variable then processing continues with steps 823 through 830.

30 In step 823 the function determines the category of the template variable. In step 824 the function determines whether unprocessed source terms 214 of the determined category type remain in the product glossary 212. If all source terms 214 of the determined 35 category type have been processed then the function continues processing with step 818.

If unprocessed source terms 214 of the determined category type remain in the product glossary 212 then in step 825 the function retrieves the first unprocessed source term 214 of the determined category type from the product glossary 212. In step 826 the function compares the retrieved source term 214 from the product glossary 212 with the source term(s) 203 in the source language document 202 located after the position indicator. In step 827 the function determines whether there was a match. If there was not a match then in step 828 the function assigns the value false to the boolean variable match. Upon completion of step 828 processing continues with step 824.

Returning to the discussion of step 827, if the function determines that there was a match then in step 829 the function assigns the value true to the boolean variable match. In step 830 the function moves the position indicator to the end of the matched source term 203 in the source language document 202. Upon completion of step 830 processing continues with step 818.

Returning to the discussion of step 822, if the function determines that the unprocessed portion of the template string is not a template variable, then processing continues with steps 831-835. In step 831 the function retrieves character strings from the template string until a template variable is reached or until the end of the template string is reached. In step 832 the function compares the retrieved character strings with the source terms 203 from the source language document 202 located after the position indicator. During the process of comparing, the function, in step 832, moves the position indicator each time a new source term is compared.

In step 833 the function determines whether there was a match during the comparison performed in step 832. If there was a match then in step 834 the function assigns the value true to the boolean variable

match. Upon completion of step 834 processing continues with step 818.

Returning to the discussion of step 833, if the function determines that there was not a match during the 5 comparison performed in step 832, then in step 835 the function assigns the value false to the boolean variable match. Upon completion of step 835 processing continues with step 818.

Figure 9 is a flow diagram of the function 10 GenerateTranslationHandle which associates a translation handle with the matched source term 203 from the source language document 202 and then associates the target term 218 from the product glossary 212, which translates the match source term 203 into the target language, with 15 the translation handle. In the preferred embodiment the translation handle generator 232 (Figure 2) performs the function GenerateTranslationHandle.

In step 901 of Figure 9 the function determines if source terms 203 from the source language document 202, 20 which matched the source terms 214 from the product glossary 212, still remain unprocessed. If unprocessed source terms 203 still remain then the function processes steps 902-905.

In step 902 the function 25 GenerateTranslationHandle retrieves the first unprocessed source term 203 from the source language document 202. In step 903 the function underlines the retrieved source term 203 and inserts a translation handle in the form of a superscript character at the end of the retrieved source 30 term 203. In the preferred embodiment, the superscript characters are cycled on a per paragraph basis. For example, assume the superscript characters J, K, L, . . . are used as the translation handles for source terms 203. If three source terms 203 are to be assigned translation 35 handles in a first paragraph, then the three source terms are assigned the superscript characters J, K, and L, respectively. If two source terms 203 in a second

paragraph are to be assigned translation handles, then the superscript characters J and K are "cycled" and assigned to the two source terms 203 in the second paragraph.

In step 904 the function retrieves the paragraph 5 tag for the paragraph in the source language document 202 which contains the retrieved source term 203. In step 905 the function stores the retrieved paragraph tag, the superscript character, the retrieved source term 203, and the target term 218, which translates the retrieved source 10 term 203 into the target language, in an index file 248 associated with the source language document 202. Upon completion of step 905 processing continues with step 901.

Returning to the discussion of step 901, if the function determines that all source terms 203 from the 15 source language document 202 have been processed, then in step 906 the function GenerationTranslationHandle returns processing control to the source language document preparer 226.

Figure 10 is a flow diagram of the function 20 InsertTranslation which inserts the target term 256 from the index file 248 associated with the source term 235 displayed on the display device 234. In the preferred embodiment, the translation inserter 236 (Figure 2) performs the function InsertTranslation.

In step 1001 of Figure 10 the function receives 25 an insert translation request. In a preferred embodiment the insert translation request is performed by pressing a key combination on the keyboard 207 (Figure 2) containing the translation handle in the form of the superscript character. In step 1002 the function retrieves the 30 translation handle in the form of the superscript character from the received insert translation request. In step 1003 the function retrieves the paragraph tag from the paragraph in the source language document 202 which 35 contains the current insertion point displayed on the display device 234. In step 1004 the function retrieves and opens the index file 248 associated with the source

language document 202. In step 1005 the function retrieves from the index file 248 the target term 256 associated with the retrieved superscript character and the retrieved paragraph tag. In step 1006 the function 5 InsertTranslation inserts both a bookmark and the retrieved target term 256 at the current insertion point on the display device 234. The bookmark marks a location in the source document 202. In step 1007 the function InsertTranslation returns processing control to the word 10 processing program 204.

Figure 11 is a flow diagram of the function ToggleFirstCharacter. In the preferred embodiment the character changer 240 (Figure 2) implements the function ToggleFirstCharacter.

15 In step 1101 of Figure 11 the function receives a toggle capitalization request. In step 1102 the function determines the location of the last bookmark inserted with an insert translation request in the source language document 202 on the display device 234. In 20 step 1103 the function retrieves the first character of a character string after the bookmark. In step 1104 the function determines if the retrieved character is a letter. If the retrieved character is a letter then in step 1105 the function determines if the retrieved first 25 character is capitalized. If the retrieved first character is capitalized then the function ToggleFirstCharacter replaces the first character with a lowercase version of the same character in step 1106. In step 1107 the function ToggleFirstCharacter returns 30 processing control to the word processing program 204.

Returning to the discussion of step 1105, if the retrieved first character is not capitalized then in step 1108 the function replaces the first character with a capitalized version of the same character. In step 1107 35 the function ToggleFirstCharacter returns processing control to the word processing program 204.

Returning to the discussion of step 1104, if the function determines that the retrieved character is not a letter then in step 1107 the function ToggleFirstCharacter returns processing control to the word processing program 204.

Figure 12 is a flow diagram of the function UndoTranslation which deletes the last target term 256 inserted in the source language document 202 on the display device 234. In the preferred embodiment, the 10 translation remover 242 performs the function UndoTranslation.

In step 1201 of Figure 12 the function receives an undo translation handle request. In step 1202 the function locates the last bookmark inserted in the source 15 language document 202 with an insert translation request. In step 1203 the function deletes all source terms 235 from the display device 234 stored between the located bookmark and the current insertion point on the display device 234. In step 1204 the function UndoTranslation 20 returns to the word processing program 204.

Figure 13 is a flow diagram of the function TranslationParagraph which either creates a new translation paragraph on the display device 234 in which to translate a source language sentence 237 or which 25 deletes a paragraph which contains source terms 235 that have been translated. In the preferred embodiment the translation paragraph creator 238 (Figure 2) performs the steps of the function TranslationParagraph.

In step 1301 of Figure 13 the function receives 30 a paragraph command. In step 1302 the function locates the source language paragraph in the source language document 202 displayed on the display device 234 which contains the current insertion point. In step 1303 the function determines if the located paragraph is preceded 35 by a colon. If the function determines that the paragraph is not preceded by the colon then in step 1304 the function inserts a colon immediately before the source

language paragraph 237 displayed on the display device 234. In step 1305 the function inserts a temporary paragraph mark to mark the end of the translation paragraph. In step 1306 the function positions the 5 insertion point just before the temporary paragraph mark. In step 1307 the function TranslationParagraph returns processing control to the translation assister 208 (Figure 2).

Returning to the discussion of step 1303, if the 10 function determines that the located paragraph is preceded by a colon, then in step 1308 the function deletes the colon, the source language paragraph immediately following the colon, and the temporary paragraph mark. Upon completion of step 1308, the function TranslationParagraph 15 returns processing control to the translation assister 208 (Figure 2).

Specific Example:

A specific example using Figures 14-21 will help 20 illustrate the preferred function of translating the source language document 202 (Figure 2) to the target language document 206. Figures 14-20 are diagrams depicting a state of the display device 234 (Figure 2) at various stages of the translation process. Figure 21 is a 25 block diagram of the product glossary 212 used to implement the translations depicted in Figures 14-20.

Figure 14A is a block diagram of the source language document 202 after the source language document preparer 226 has prepared it for translation by the user 30 of the system 200. The source language document 202 includes the translation handle 1402 and the insertion point 1404.

Figure 14B depicts a state of the display device 234 after the German translation, "Text in einem," is 35 inserted at the location of the current insertion point 1404. The user manually types in the translation because the phrase "Finding Text in a" does not have a translation

handle associated with it. When the user of a system 200 is ready to translate the English language word "document" to the German language, the user enters the insert translation request, CTRL+SHIFT+J, which inserts the 5 German translation (i.e., the target term 256 from the index file 248) for the English word Document (i.e., the source term 203) at the location of the current insertion point 1404. Figure 15A depicts a state of the display device 234 after the user entered the insertion request 10 for "document", received the translation (i.e. "Document"), and manually typed in the translation for "find" (i.e. "finden").

Now that the heading 1504, "Finding Text in a document", has been completely translated, the user 15 deletes the heading 1504. Figure 15B depicts a state of the display device 234 after the user deletes the heading 1504.

In Figure 16A the sentence 1602 has been automatically inserted in the source language document 202 20 by the source language document preparer 226. In the preferred embodiment of the present invention, when a template string which comprises a complete sentence is found in the source language document 202, the source language document preparer 226 of the translation assister 25 208 inserts the target term 256 from the index file 248 associated with the template string into the source language document 202. As Figure 21 depicts, the German phrase "Wählen Sie aus dem Menu %MNU den Befehl %COM" is the target term 256 for the template string "From the %MNU 30 menu choose %COM".

The next line in the source language document 202 contains the template string "Find What box" (see Figure 21) associated with the translation handle 1606. The user proceeds to translate the sentence "In the Find 35 What box, type the text you want to find". More specifically, the user proceeds to translate that sentence in a translation paragraph. Therefore, the user enters

the create translation paragraph command which inserts a temporary paragraph marker to mark the end of the translation paragraph and a colon 1608 immediately in front of the English language text to be translated. The 5 user first manually types in the translation for "In the" (i.e. "In dem"). The user then enters using the keyboard 207 the insert translation request for the translation handle "J" and, as Figure 16B depicts, the German translation "Zu findender Text" (i.e., the associated 10 target term 256) is inserted in the translation paragraph by the translation inserter 236. It should be noted that the translation handle "J" has been "cycled" in the sense that the translation handle "J" was used earlier to insert the target term 256 associated with the source term 15 "Document" found in an earlier paragraph.

Figure 17A depicts what a state of the display device 234 would have looked like had the user of the system 200 mistakenly entered the insert translation 20 request for the translation handle "K" instead of the insert translation request for the translation handle "J". The display screen 234 would have appeared in the state of Figure 17A with the phrase "Wahlen Sie dann die Schaltflache Weitersuchen" inserted in the source language 25 document 202, instead of in the state of Figure 16B, with the phrase "Zu findender Text" inserted in the source language document 202. Oftentimes, the user of the system 200 will continue typing on the keyboard 207 before recognizing that an inappropriate target term 256 has been 30 inserted into the source language document 202. Figure 17A depicts such a case, with the phrase "geben Sie" entered after the mistaken insertion of the target term 256. In such a case, the user of the system 200 merely invokes the undo translation handle request, which in the 35 preferred embodiment comprises CTRL+SHIFT+X. Figure 16B depicts a state of the display device 234 after the undo translation handle request has been entered.

Turning to Figures 18A and 18B, the user proceeds to translate the sentence "Then choose the Find Next button", using the translation handle 1804. As Figure 21 depicts, the phrase "Then choose the Find Next button" is a template string.

Figure 18B depicts a state of the display screen 234 after the user has entered the insert translation request for the translation handle "K". However, the first character of the target term 256 associated with the phrase "Then choose the Find Next button" contains inappropriate capitalization given its position within the translated sentence, namely a capital "W" for the term "Wahlen". The user then enters the toggle first character command which comprises the entry of CTRL+SHIFT+0 on the keyboard 244. Note that the user enters the toggle first character command to change the first character 1806 while the current insertion point 1808 is not located at the first character 1806. Figure 19A depicts a state of the display device 234 after the user enters the toggle first character command and the character changer 240 has inserted the character 1902, in the form of a lowercase "w", in place of the character 1806 (Figure 18B) which contained the capital "W".

Figure 19A also depicts the English paragraph  
25 In the Find What box, type the text  
you want to find. Then choose the  
Find Next button to begin the search.

which has been completely translated. By entering the paragraph command a second time, the user of the system 30 200 deletes the English paragraph. Figure 20A depicts a state of the display screen 234 after the translation paragraph creator 238 has deleted the translated English paragraph.

Figure 20B depicts a state of the display screen 35 234 after the remaining portion of the document 202 has been translated using the techniques set forth in more

detail above, thus creating the target language document 206.

It will be appreciated that, although a specific embodiment of the invention has been described herein for 5 purposes of illustration, various modifications may be made without departing from the spirit and scope of the invention. Accordingly, the invention is not limited except as by the appended claims.

Claims

1. A computer system for translating a source language document written in a source language to a target language document written in a target language, the source language including a multiplicity of source terms and the target language including a multiplicity of target terms, the computer system including a display screen, the source language document, a product glossary having a plurality of source terms from the source language and a plurality of target terms from the target language, each source term being associated with the corresponding target term which translates the source term into the target language, the computer system comprising:

means for producing a translation screen portion on the display screen, the translation screen portion including a current insertion point;

means for displaying the source language document on the translation screen portion;

means for comparing each of the plurality of source terms from the product glossary with the source terms in the source language document;

inserting means for inserting a character adjacent to the source term in the source language document, in response to each comparison by the comparing means which produces a match between one of the source terms in the source language document and one of the source terms in the product glossary;

means for associating in an index file the inserted character with a target term from the product glossary that translates the matched source term from the source language into the target language;

means for inputting an insert target term command which contains a translation request character corresponding to the inserted character;

means for retrieving the translation request character from the insert target term command;

means for retrieving from the index file the target term associated with the retrieved translation request character; and

means for inserting the retrieved target term on the translation screen portion in response to the insert target term command.

2. The computer system of claim 1 further comprising means for underlining the matched source term in response to each comparison by the comparing means which produces a match between one of the source terms in the source language document and one of the source terms in the product glossary.

3. The computer system of claim 1 further comprising means for replacing a lowercase version of a first character in a most recent target term inserted by the insertion means, with an uppercase version of the character without having to move the current insertion point into alignment with the first character.

4. The computer system of claim 1 further comprising means for deleting a most recent target term inserted by the insertion means, without having to move the current insertion point into alignment with the most recently inserted target term.

5. The computer system of claim 1 wherein the insertion means includes means for inserting the retrieved target term at the current insertion point on the translation screen portion.

6. The computer system of claim 1 wherein the target terms include template strings which are one of a partial and a full sentence containing variables.

7. A method executed in a computer system for translating a source language document written in a source language to a target language document written in a target language, the source language including a multiplicity of source terms and the target language including a multiplicity of target terms, the computer system including a display screen, the source language document, a product glossary having a plurality of source terms and a plurality of target terms, with each source term being associated with the corresponding target term which translates the source term into the target language, the method comprising the steps of:

producing a translation screen on the display screen, the display screen including a current insertion point indicating where data will be inserted on the translation screen;

displaying the source language document on the translation screen;

comparing each of a plurality of source terms from the product glossary with the source terms in the source language document;

inserting a character immediately after the source term in the source language document displayed on the translation screen in response to each comparison in the step of comparing which produces a match between one of the source terms in the source language document and one of the source terms in the product glossary;

associating in an index file the inserted character with the corresponding target term from the product glossary that translates the matched source term from the source language into the target language;

inputting an insert command which contains a translation request character corresponding to the inserted character;

retrieving the translation request character from the insert command;

retrieving from the index file the target term associated with the retrieved translation request character; and

inserting the retrieved target term on the translation screen.

8. The method of claim 1 further comprising the step of underlining the matched source term in response to each comparison in the step of comparing which produces a match between one of the source terms in the source language document and one of the source terms in the product glossary.

9. The method of claim 1 further comprising the step of replacing a lowercase version of a first character in a most recent target term inserted in the step of inserting, with an uppercase version of the first character, without having to move the current insertion point into alignment with the first character.

10. The method of claim 1 further comprising the step of deleting a most recent target term inserted by the inserting means, without having to move the current insertion point into alignment with the target term..

11. The method of claim 1 wherein the step of inserting includes the step of inserting the retrieved target term at the current insertion point on the translation screen.

12. The method of claim 1 wherein the target terms include template strings which are one of a partial and a full sentence containing variables.

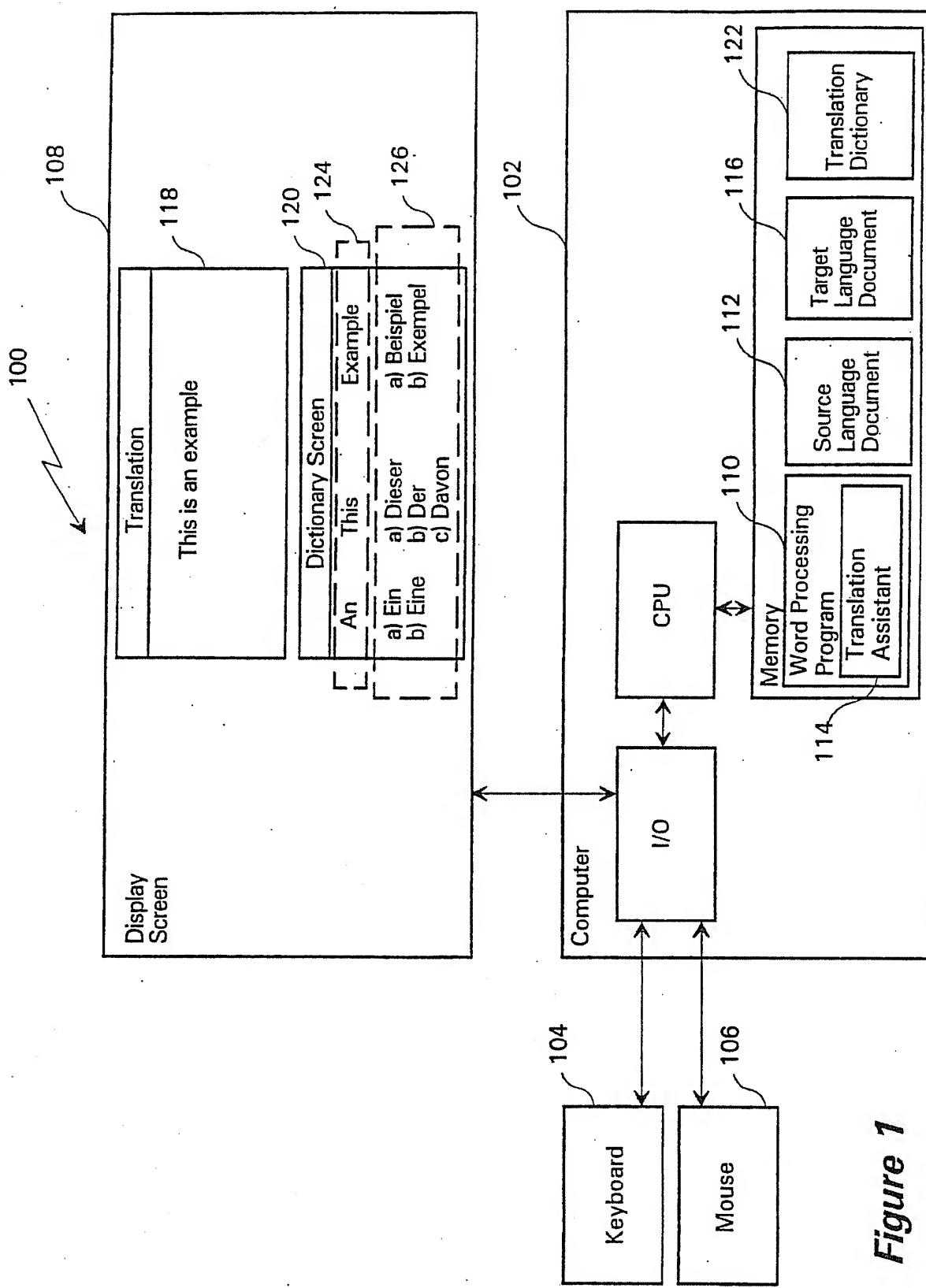
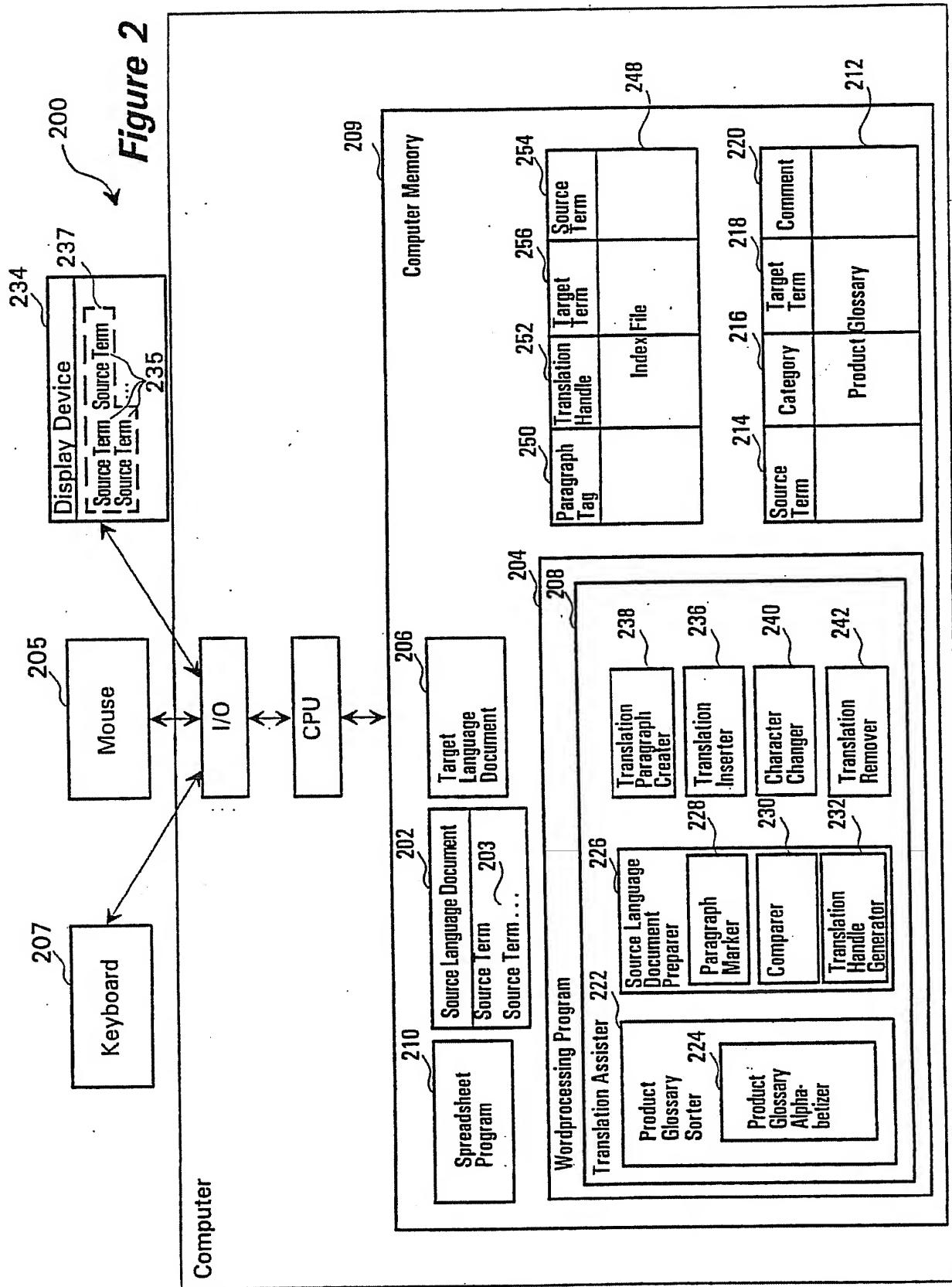
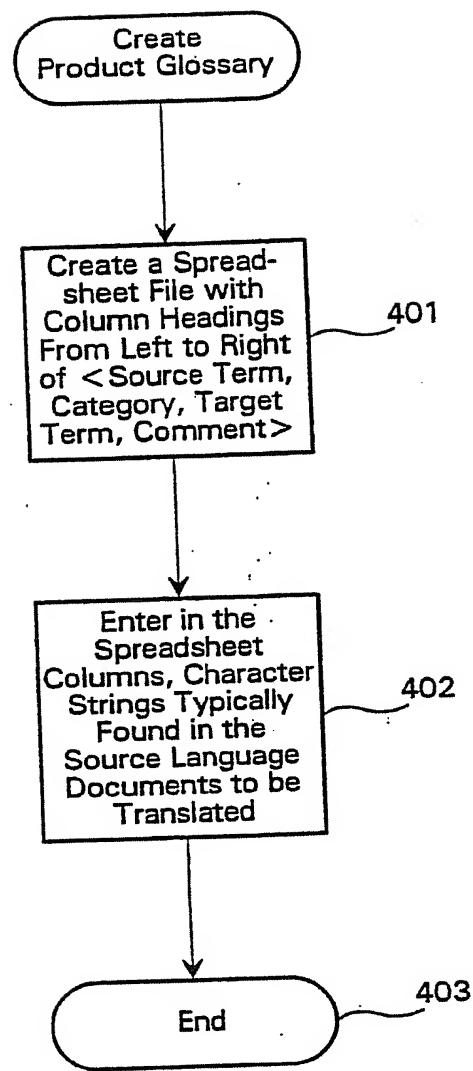


Figure 1

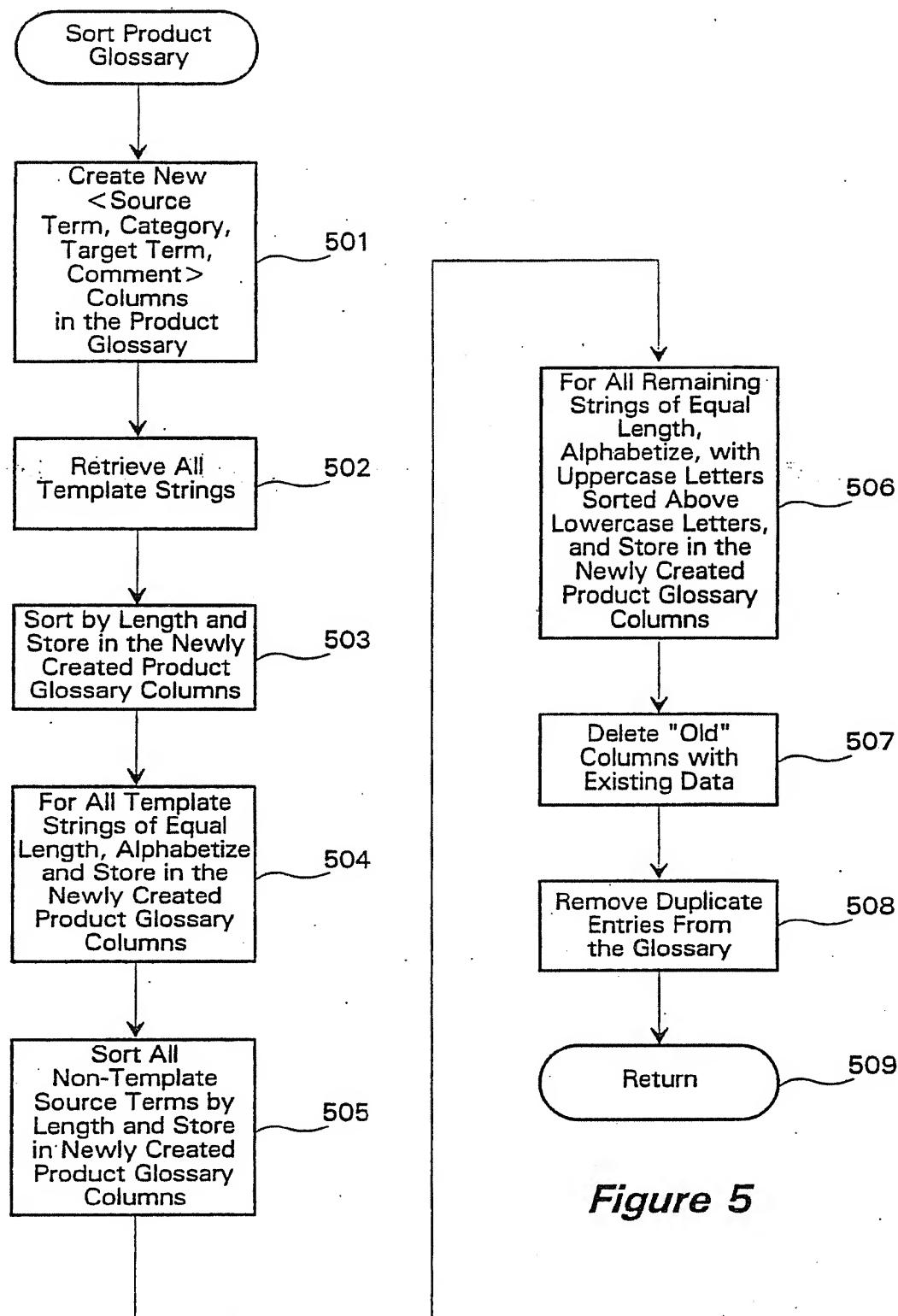


CATEGORY	DESCRIPTION	EXAMPLE
MNU	Menu name	File
COM	Command name	Print
TDB	Title of dialog box	Print
BUT	Name on a button	OK (also: Cancel, Page Setup, Help)
TXB	Text box name	From (also: To, Copies)
OPT	Option button name (round buttons in dialog boxes)	All (also: Pages, Sheet, Notes, Both)
CHX	Check box name	Preview (also: Fast, but no Graphics)
TXT	Text string in a dialog box	Printer (also: Print Range, Print)
DIA	All other dialog box items	Print Quality (this is a "combo box")
MSG	Message (i.e., status, error)	For Help on dialog settings, press F1 (appears in status bar when the Print dialog box is displayed)
MAC	Macro name	Built-in macro in Excel, Project, Word, etc.
KEY	Key name	ENTER, F1, CAPS LOCK
FLD	Field name	E.g., Date or Author fields in WinWord

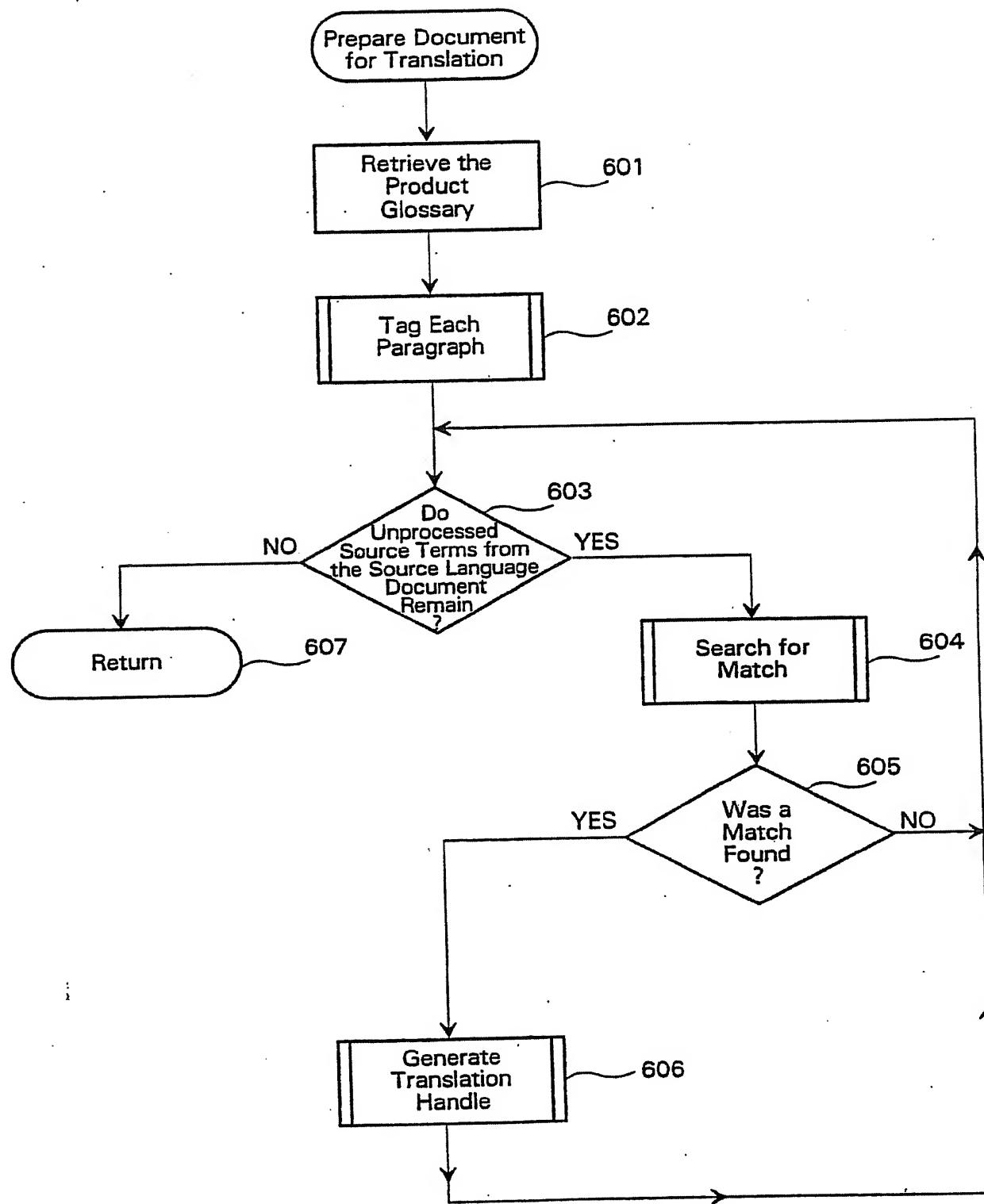
Figure 3

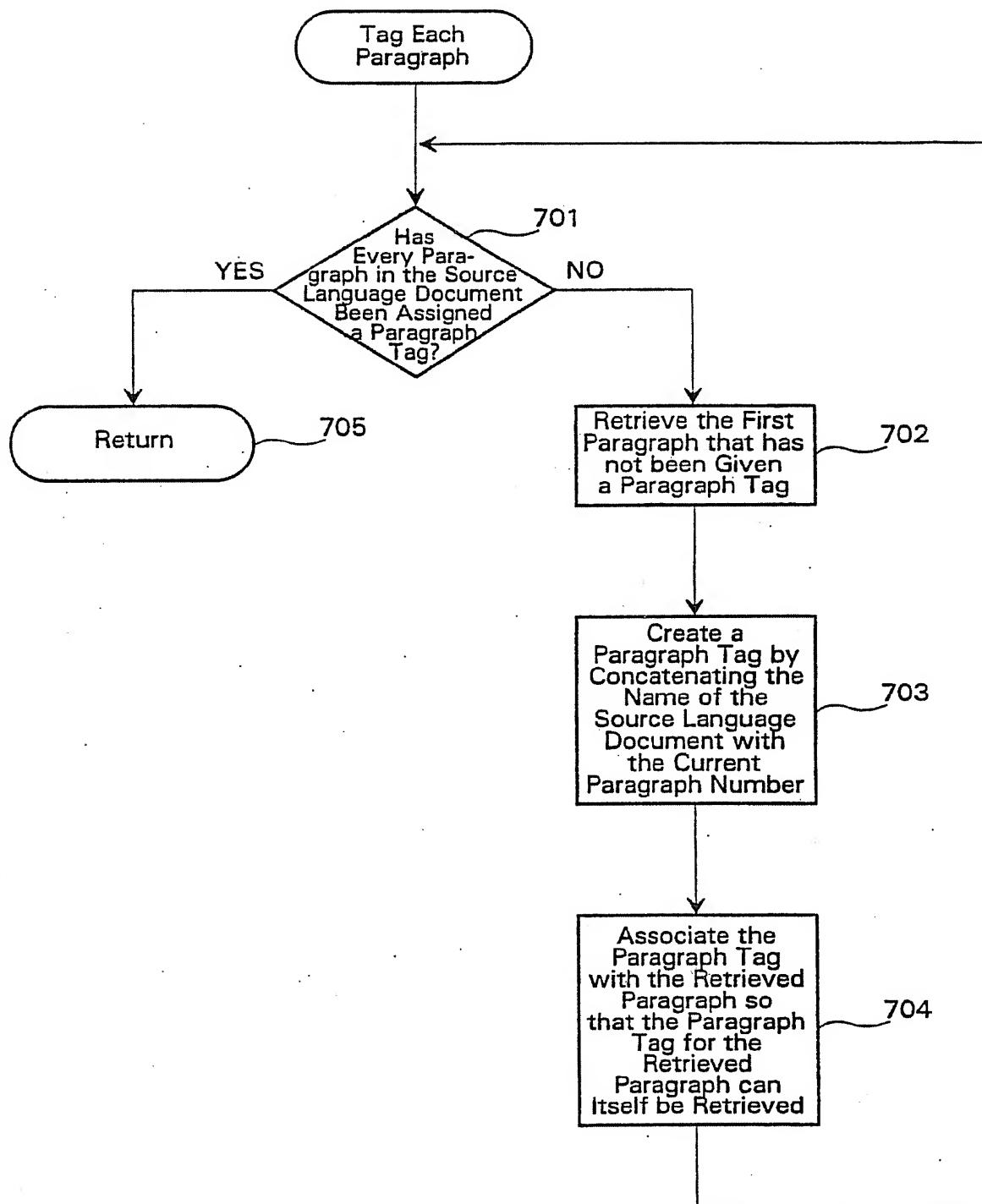


**Figure 4**

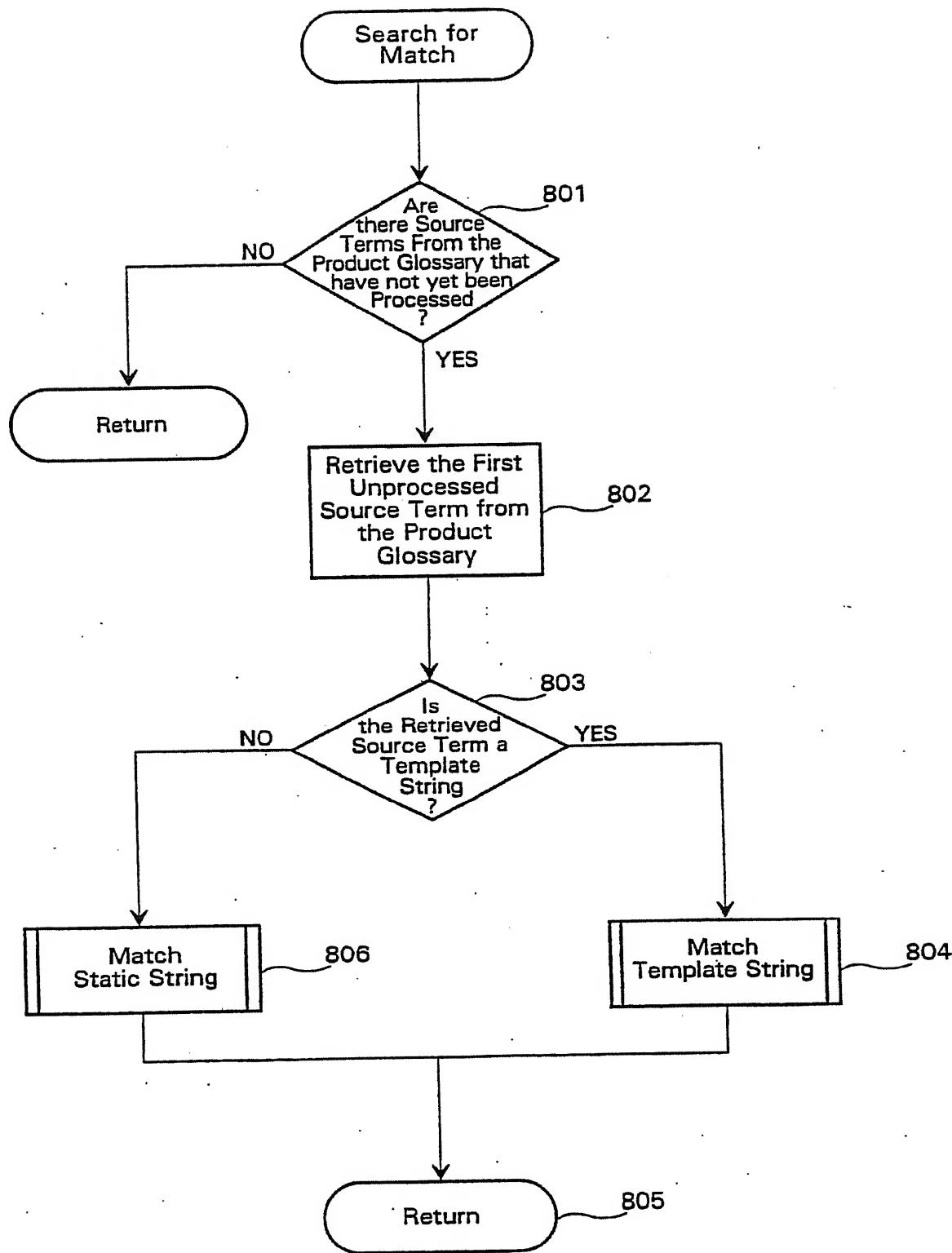
*Figure 5*

6 / 24

**Figure 6**

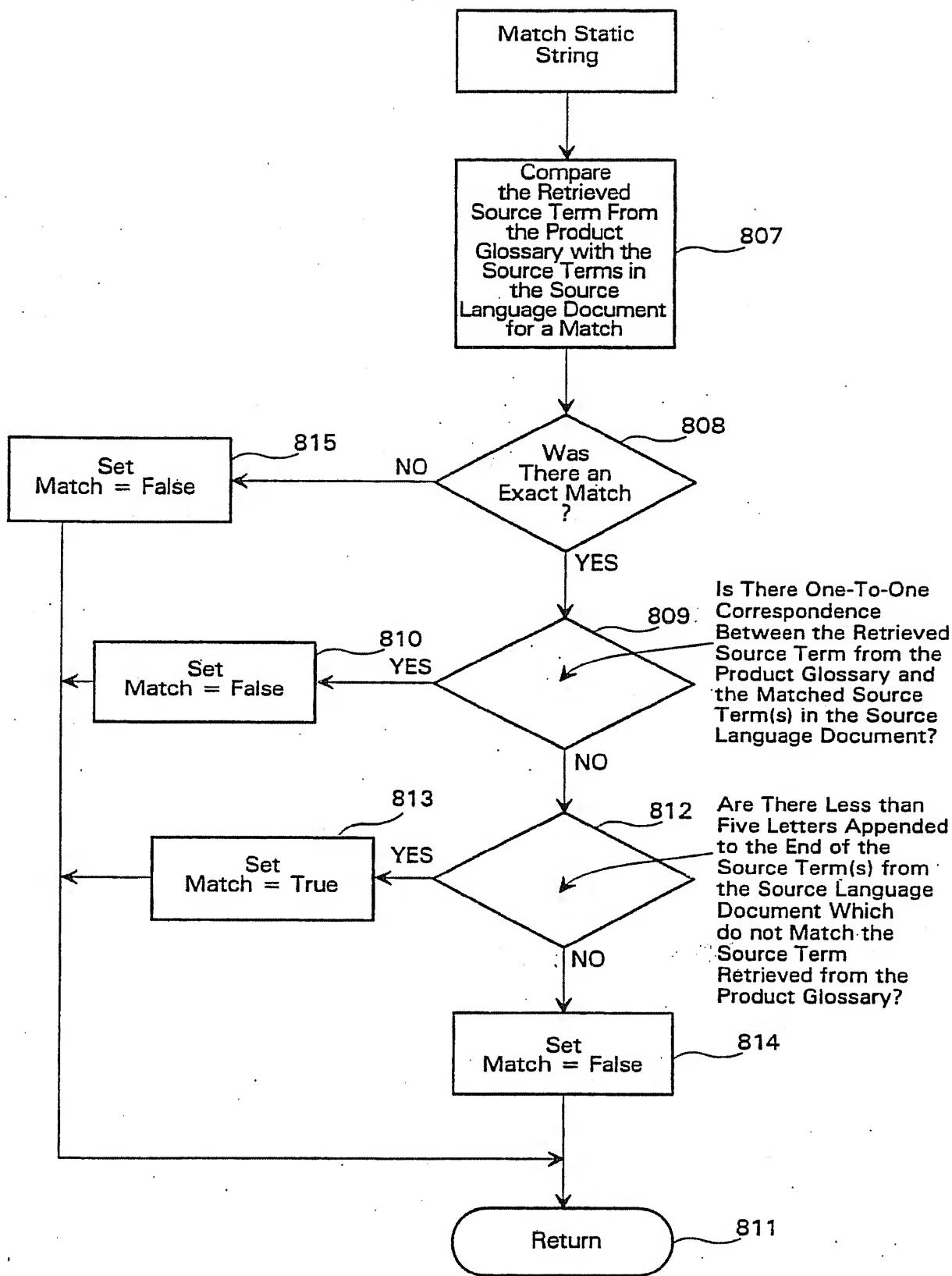


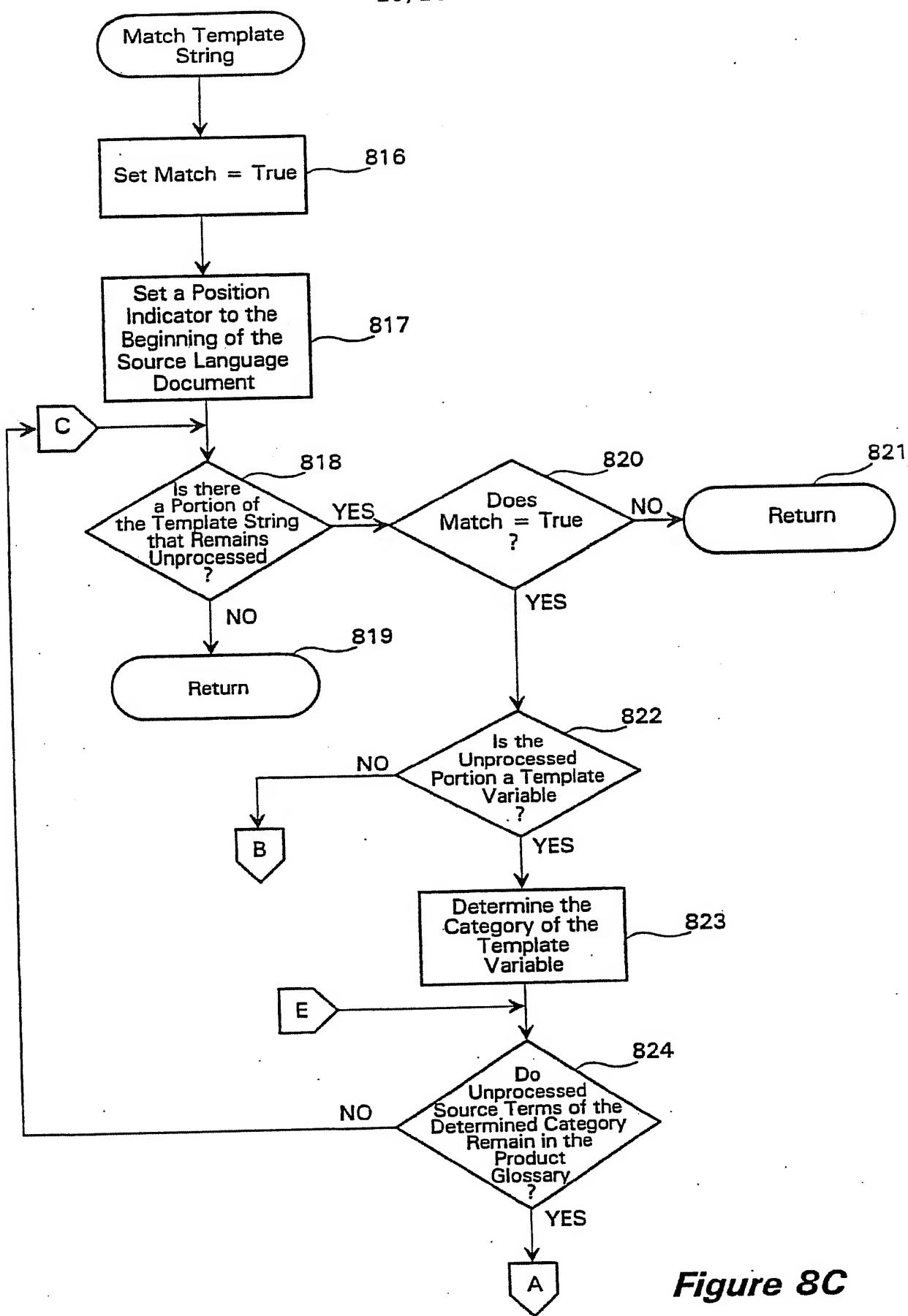
*Figure 7*

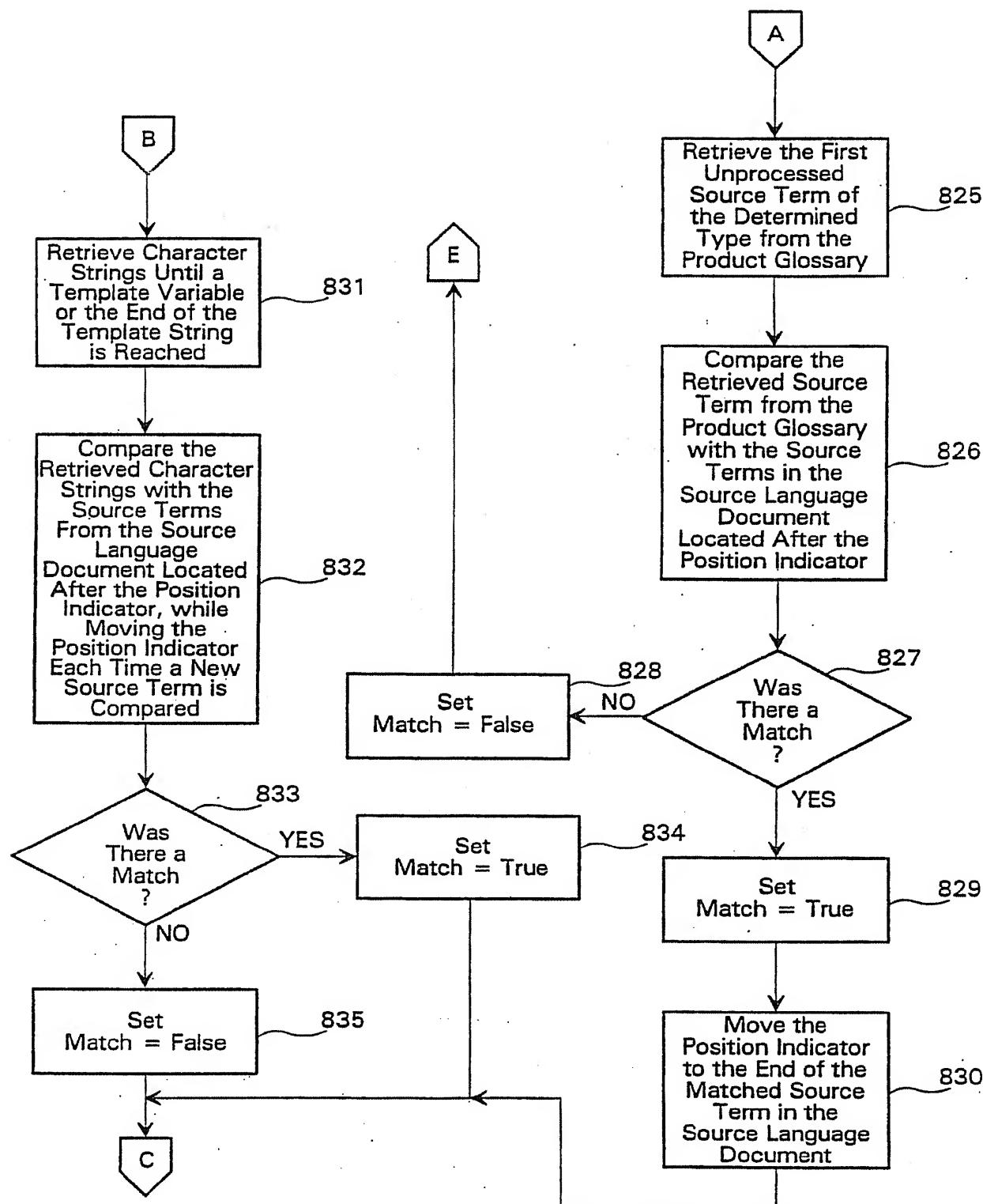


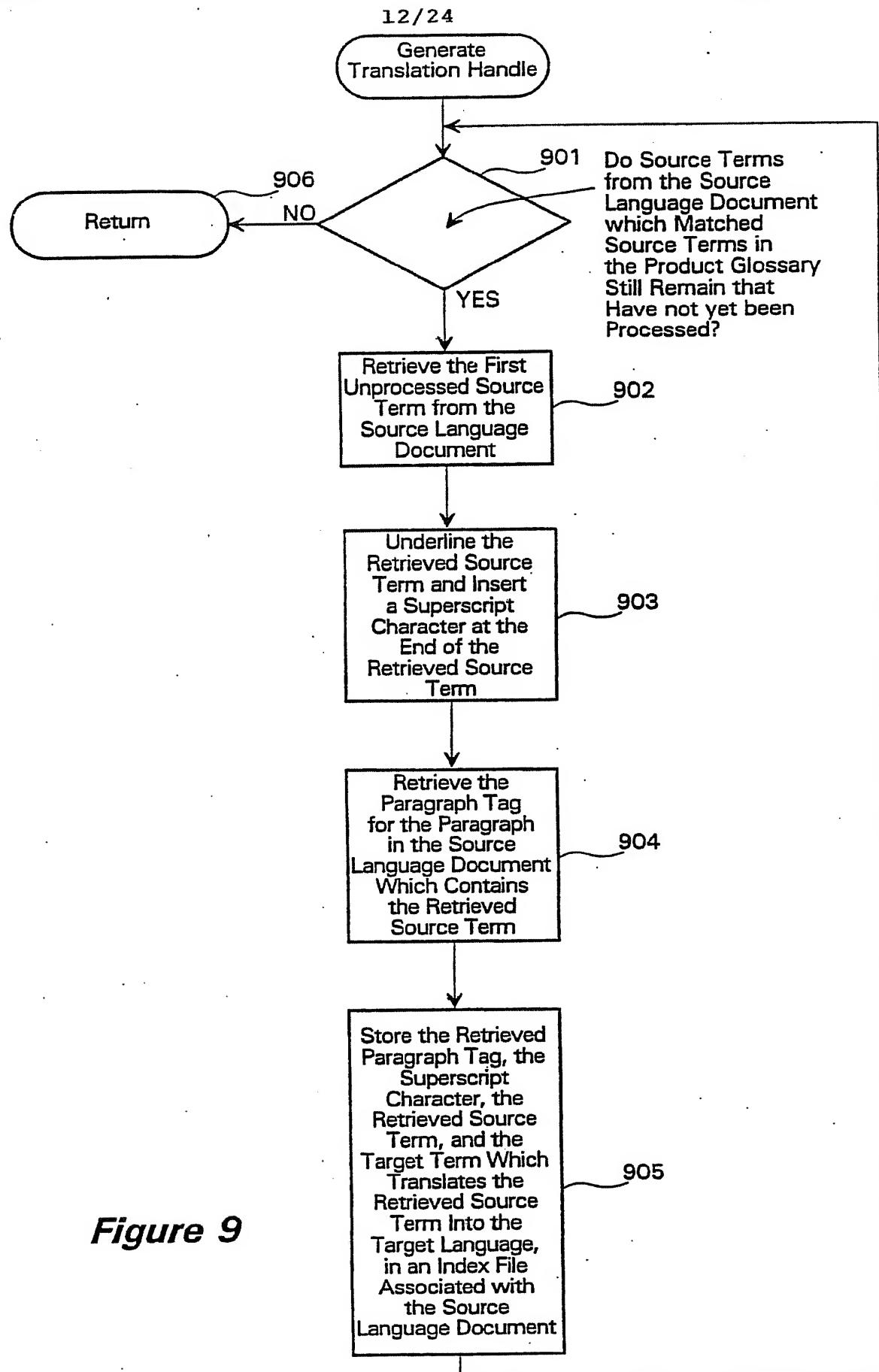
**Figure 8A**

9 / 24

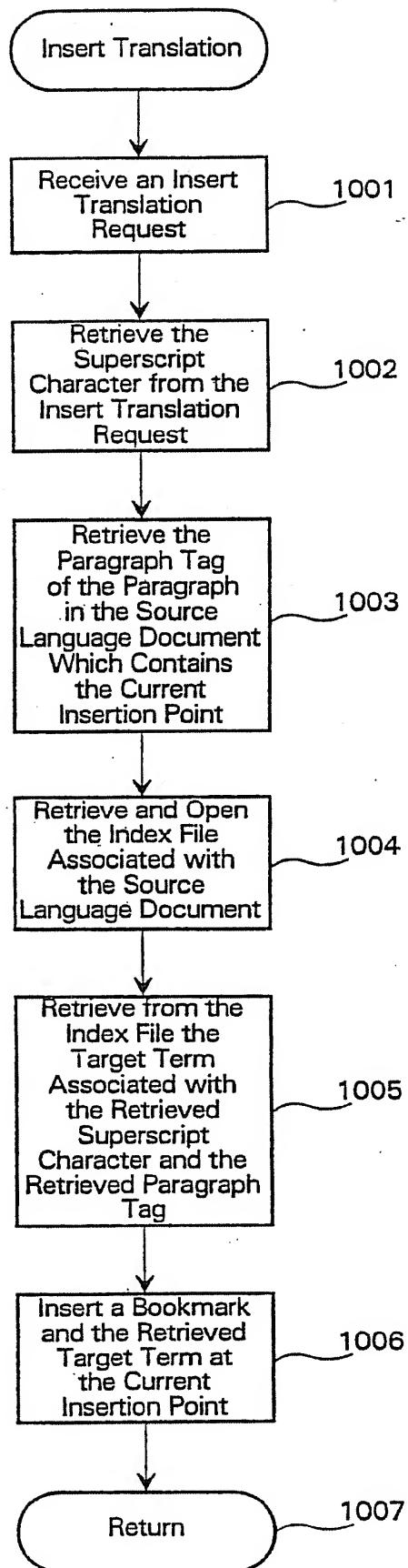
**Figure 8B**

**Figure 8C**

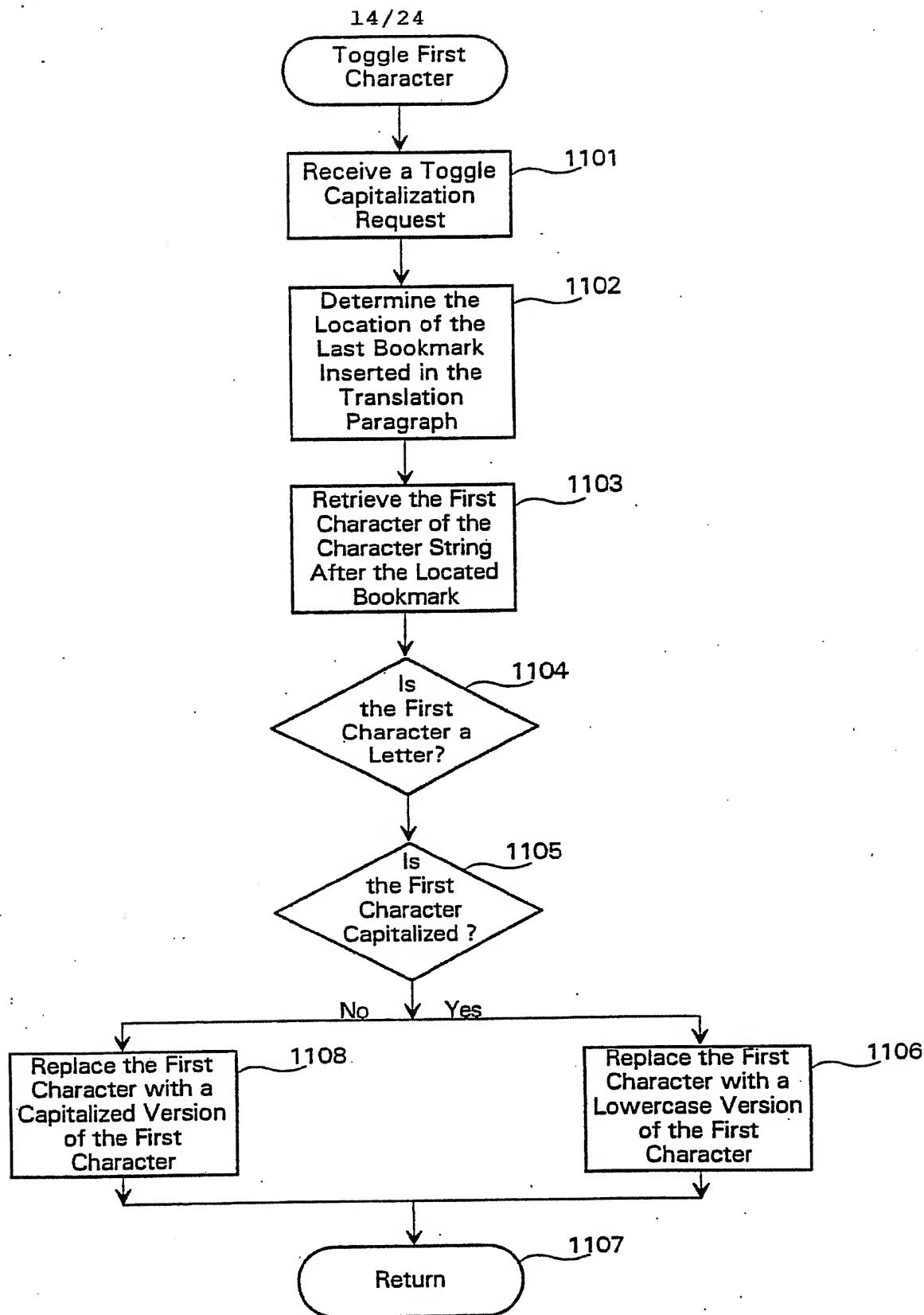
*Figure 8C (Continued)*



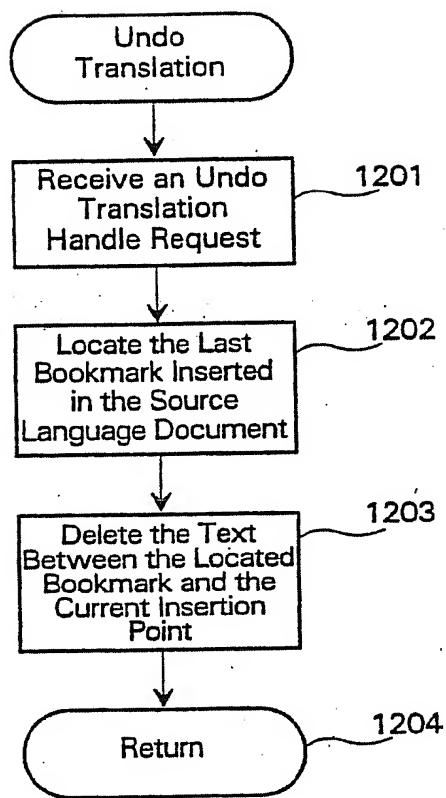
**Figure 9**



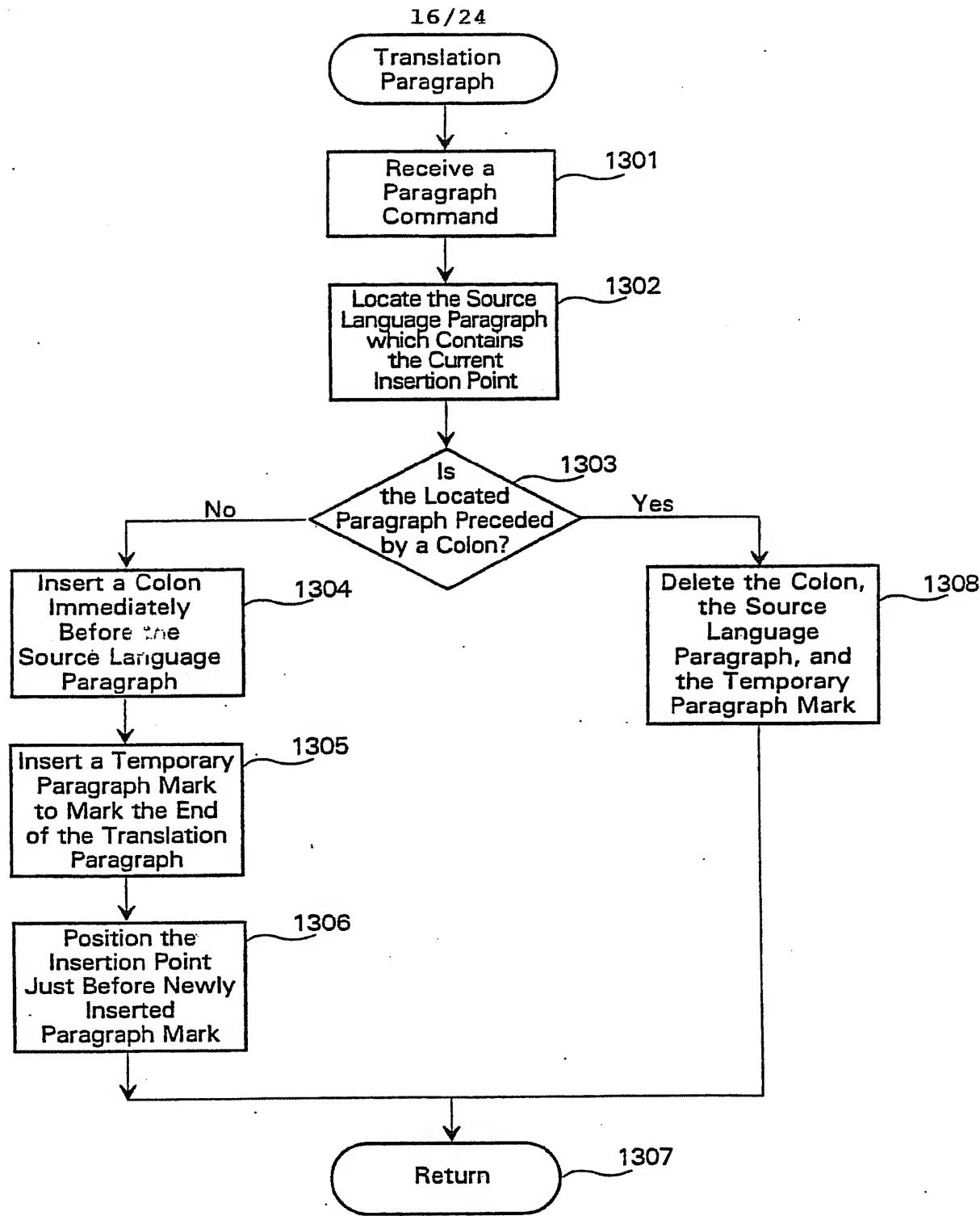
**Figure 10**



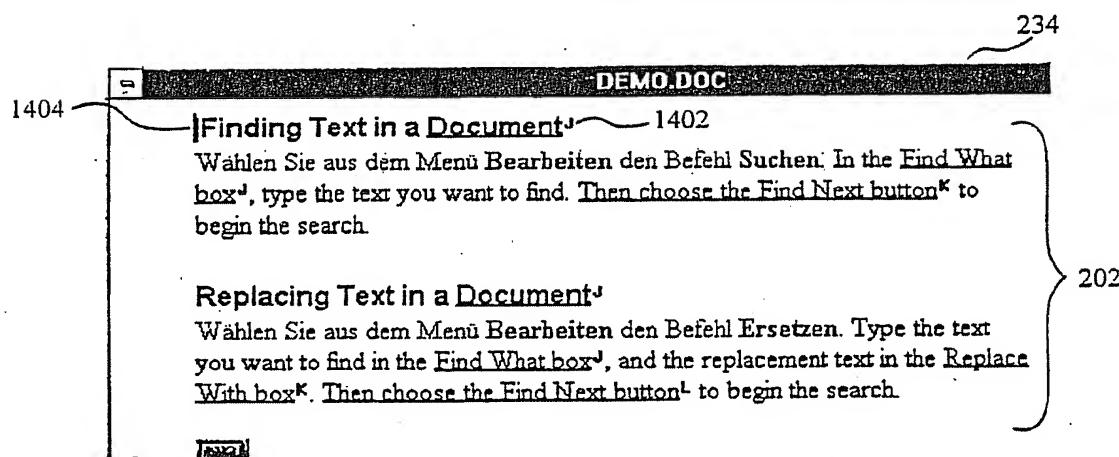
**Figure 11**



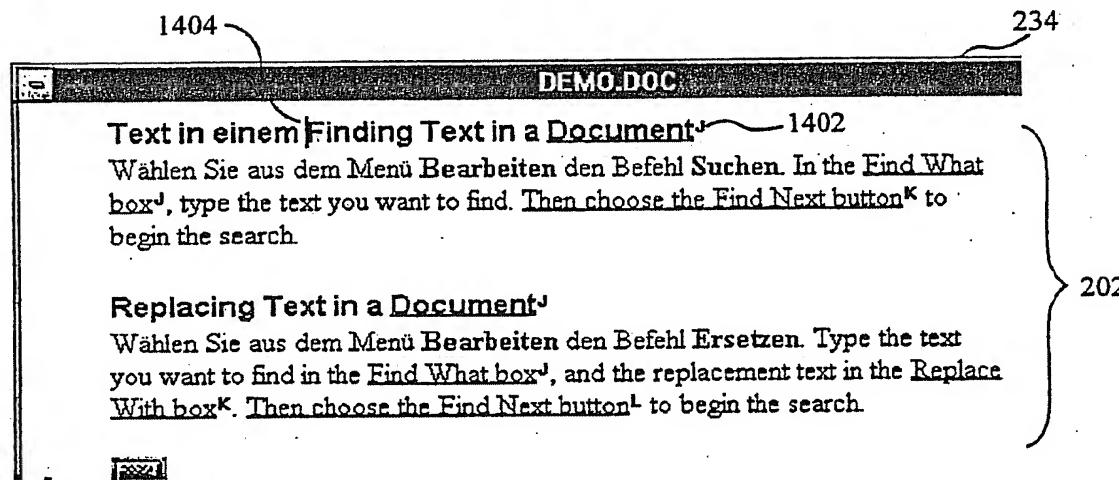
**Figure 12**



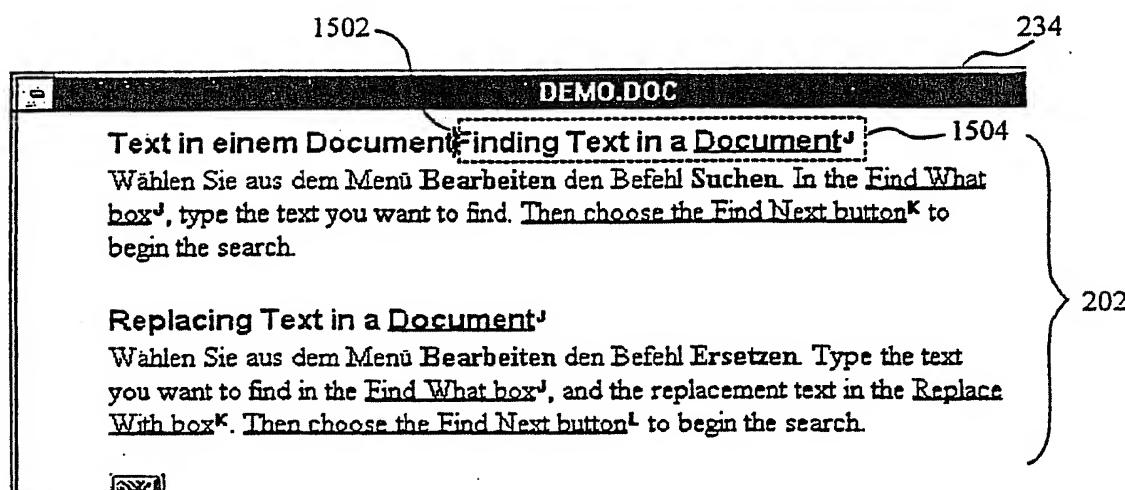
**Figure 13**



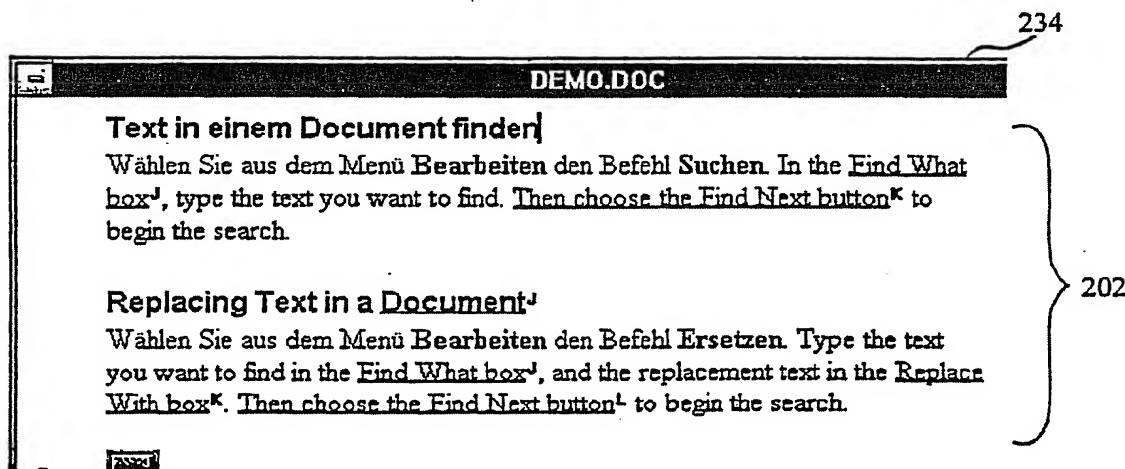
*Figure 14A*



*Figure 14B*

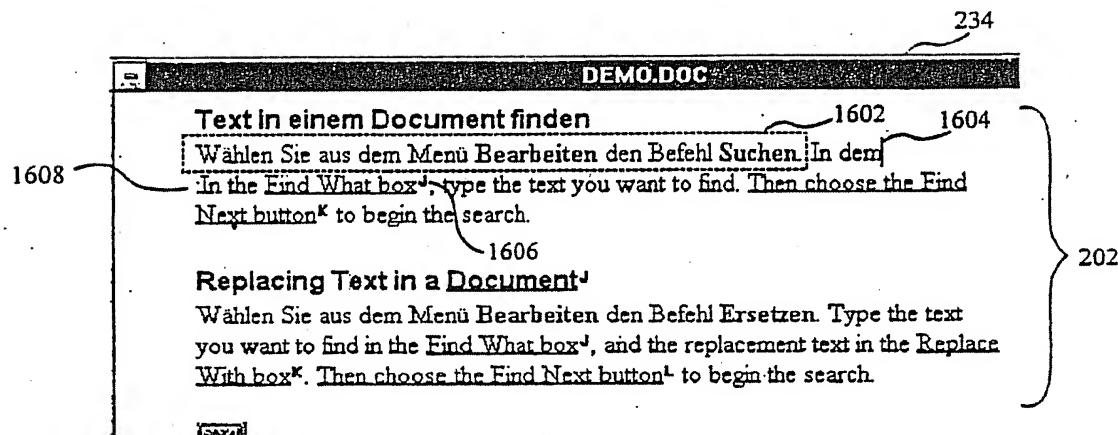
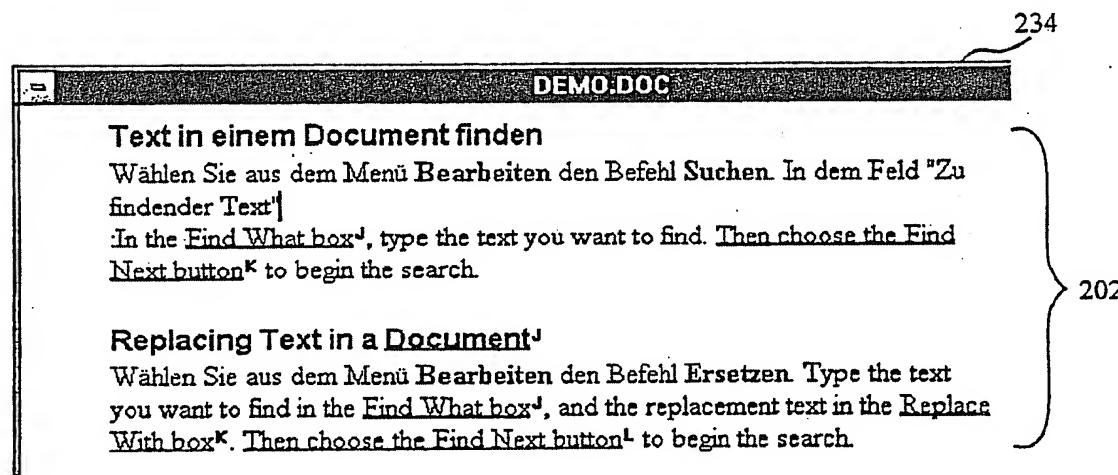


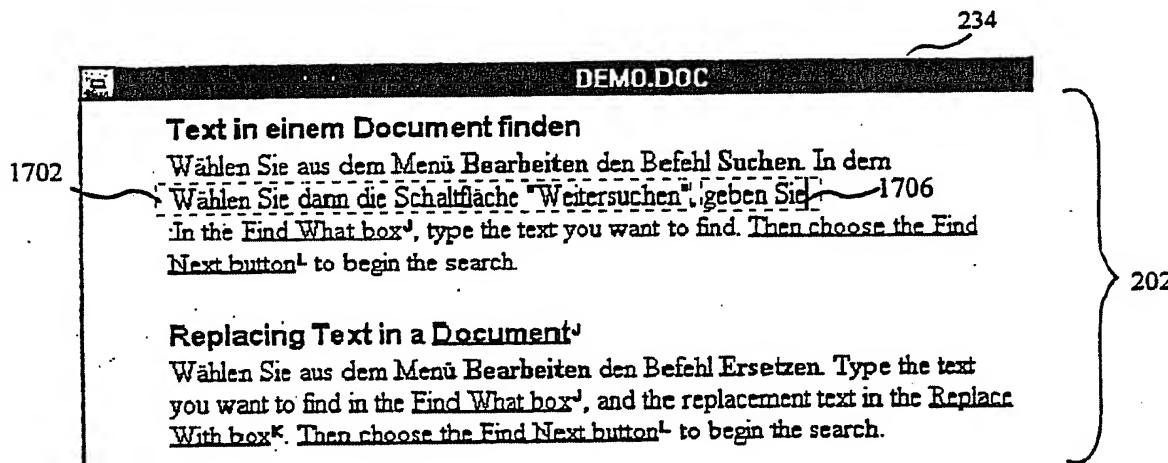
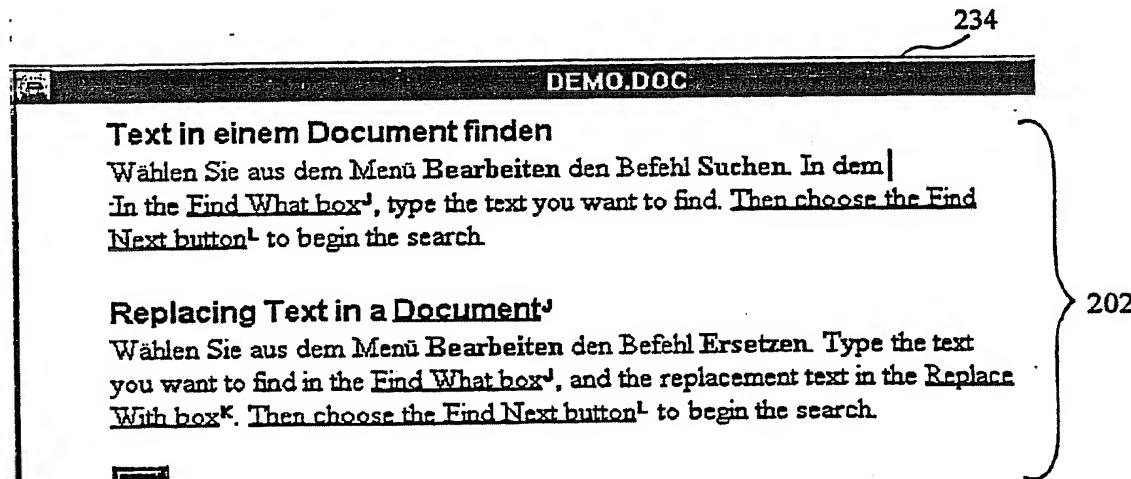
*Figure 15A*

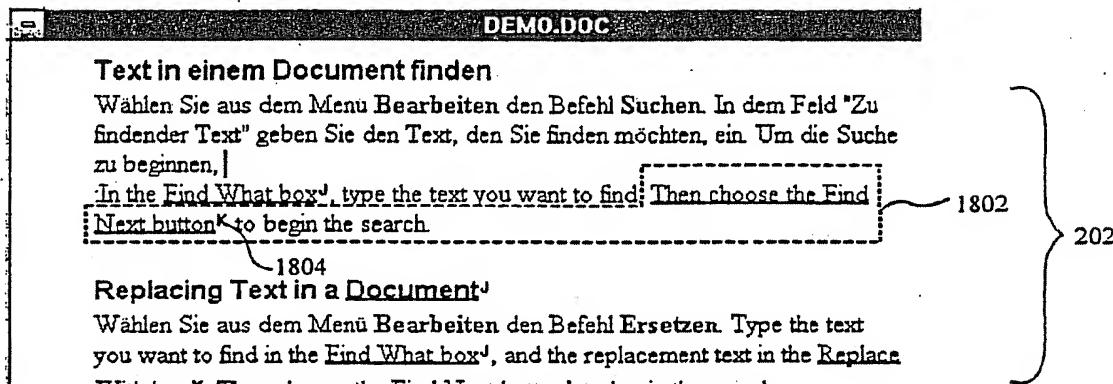
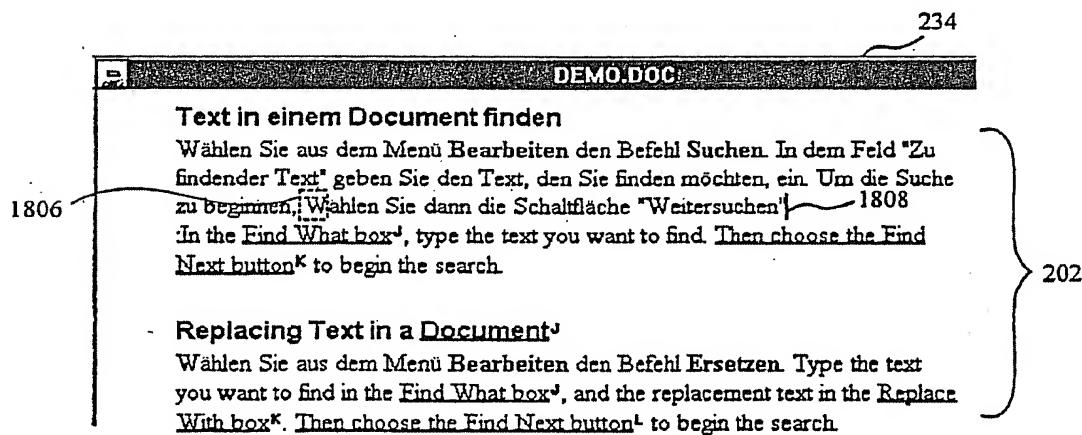


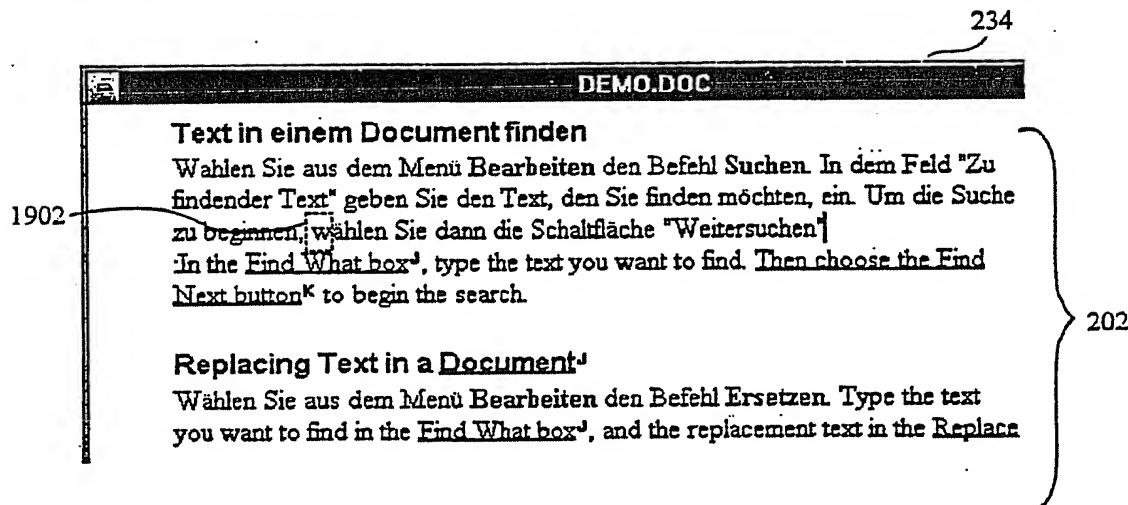
*Figure 15B*

19/24

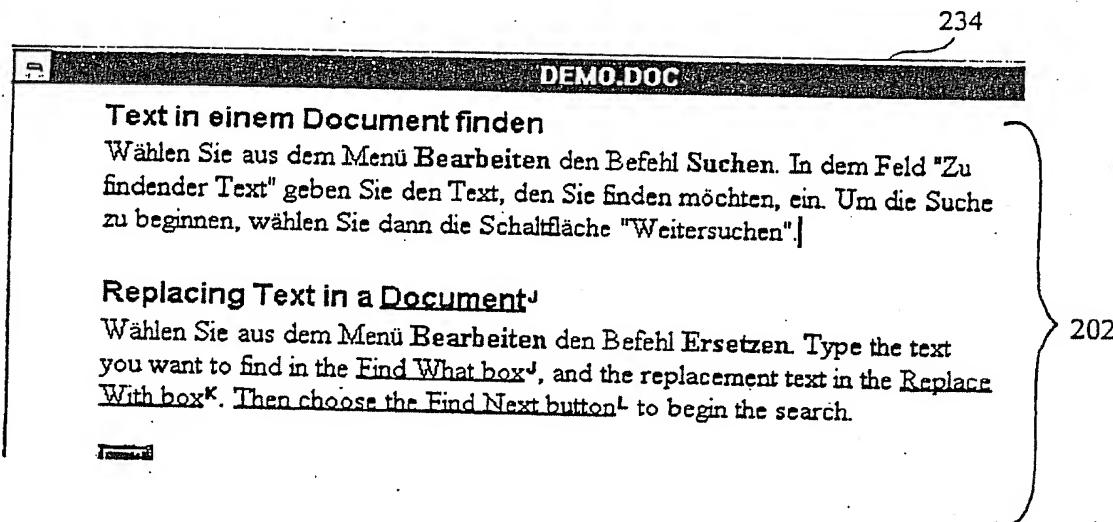
*Figure 16A**Figure 16B*

*Figure 17A**Figure 17B*

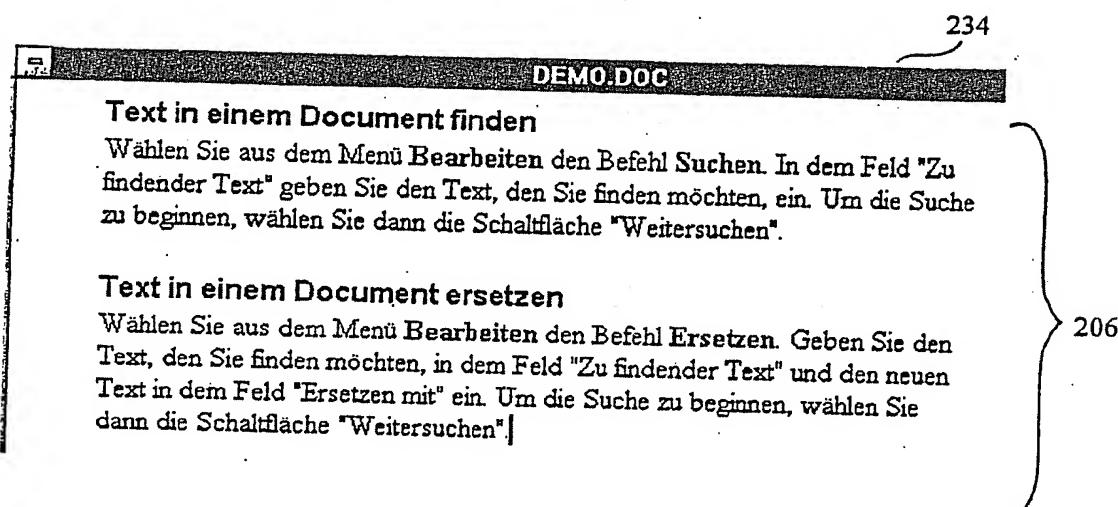
*Figure 18A**Figure 18B*



*Figure 19A*



*Figure 20A*



*Figure 20B*

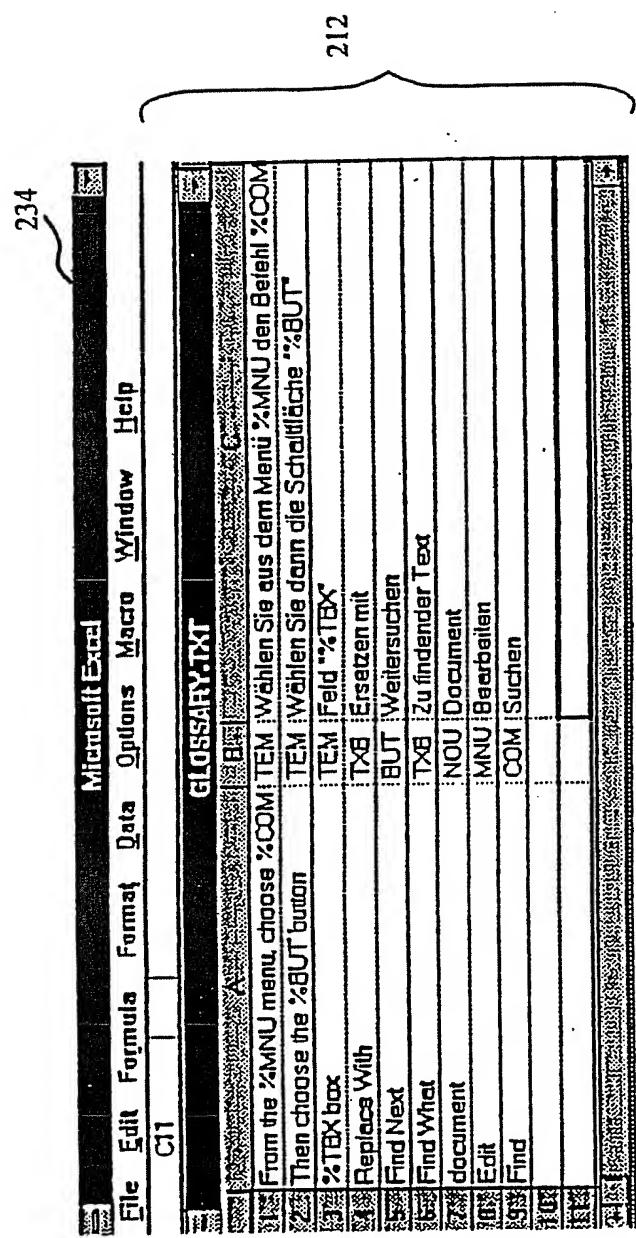


Figure 21

## INTERNATIONAL SEARCH REPORT

International Application No  
PCT/US 94/02136

A. CLASSIFICATION OF SUBJECT MATTER  
IPC 5 G06F15/38

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)  
IPC 5 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP,A,0 514 305 (IBM) 19 November 1992 see page 2, column 2, line 13 - line 38 ---	1-12
A	EP,A,0 357 344 (SHARP) 7 March 1990 see page 2, column 2, line 21 - page 3, column 3, line 56 ---	1-12
A	GB,A,2 199 170 (SHARP) 29 June 1988 see page 4, line 16 - page 6, line 9 ---	1-12
A	GB,A,2 211 641 (SHARP) 5 July 1989 see page 2, line 4 - page 4, line 11 ---	1-12
A	US,A,4 685 060 (YAMANO ET AL.) 4 August 1987 see column 1, line 36 - line 55 ---	1-12
		-/-

Further documents are listed in the continuation of box C.

Patent family members are listed in annex.

## \* Special categories of cited documents :

- \*A\* document defining the general state of the art which is not considered to be of particular relevance
- \*E\* earlier document but published on or after the international filing date
- \*L\* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- \*O\* document referring to an oral disclosure, use, exhibition or other means
- \*P\* document published prior to the international filing date but later than the priority date claimed

- \*T\* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- \*X\* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- \*Y\* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- \*&\* document member of the same patent family

1 Date of the actual completion of the international search

15 July 1994

Date of mailing of the international search report

20.07.94

## Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+ 31-70) 340-2040, Tx. 31 651 epo nl,  
Fax: (+ 31-70) 340-3016

## Authorized officer

Büro, S.P.

## INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 94/02136

## C(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	GB,A,2 213 971 (SHARP) 23 August 1989 see page 2, line 17 - page 3, line 15 ---	1-12
A	EP,A,0 366 142 (TOSHIBA) 2 May 1990 see page 2, column 2, line 40 - page 3, column 3, line 21 ---	1-12
A	COMPUTERS & THE HUMANITIES, vol.21, no.3, July 1987, USA pages 203 - 206 M. OLSEN 'Mercury: A Translator's Glossary Manager' see the whole document ---	1-12
A	FR,A,2 446 517 (SYSTRAN) 8 August 1980 see page 2, line 21 - line 29 ---	1-12
A	GB,A,2 241 359 (SHARP) 28 August 1991 see abstract ---	1-12
A	EP,A,0 370 774 (TOSHIBA) 30 May 1990 see abstract -----	1-12

**INTERNATIONAL SEARCH REPORT**

Information on patent family members

International Application No

PCT/US 94/02136

Patent document cited in search report	Publication date	Patent family member(s)		Publication date
EP-A-0514305	19-11-92	US-A-	5251130	05-10-93
		JP-A-	6028395	04-02-94
EP-A-0357344	07-03-90	JP-A-	2061763	01-03-90
GB-A-2199170	29-06-88	JP-A-	63136264	08-06-88
		JP-A-	63136265	08-06-88
		US-A-	5010486	23-04-91
GB-A-2211641	05-07-89	JP-A-	1113871	02-05-89
US-A-4685060	04-08-87	JP-A-	60089275	20-05-85
GB-A-2213971	23-08-89	JP-A-	1169577	04-07-89
EP-A-0366142	02-05-90	JP-A-	2239368	21-09-90
		JP-A-	2238566	20-09-90
		JP-A-	2118774	07-05-90
		US-A-	5140522	18-08-92
FR-A-2446517	08-08-80	NONE		
GB-A-2241359	28-08-91	JP-A-	3222067	01-10-91
		JP-A-	3222069	01-10-91
EP-A-0370774	30-05-90	JP-A-	2140868	30-05-90
		US-A-	5214583	25-05-93